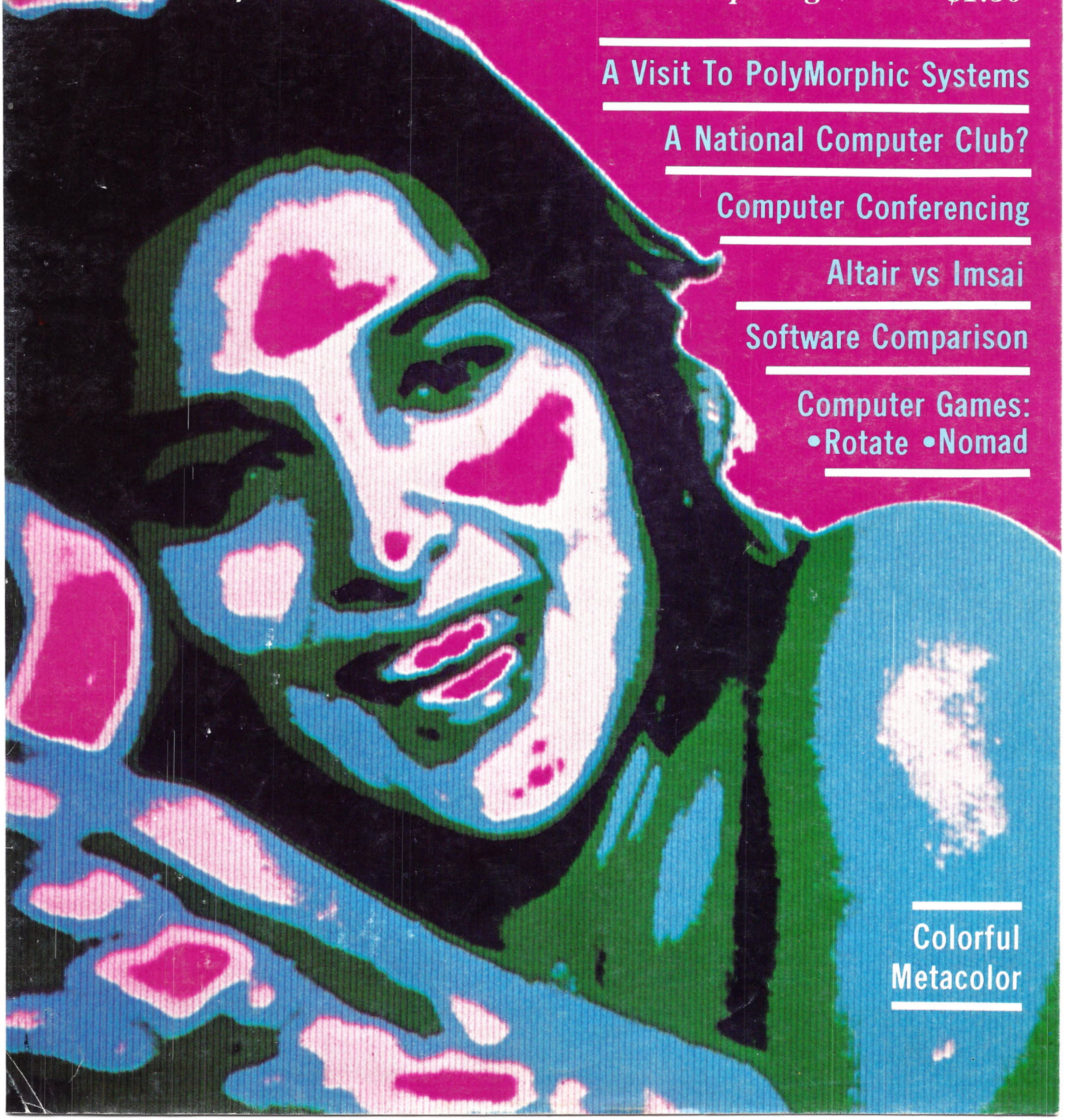
Sep-Oct 1977
vol 3, no 5

creative computing

the magazine of recreational and educational computing

Austria Sch 32.00
Belgium BF 70.00
Canada $1.50
Denmark Kr 11.00
Finland Mk 7.00
France F 9.00
Germany DM 4.50
Great Britain £ 1.00
Greece Dr. 65.00
Holland Dfl 5.00
Italy L 1500
Norway Kr 10.00
Portugal Esc 55.00
Spain Ptas 120.00
Sweden Kr 8.00
Switzerland SF 4.50
USA $1.50

$1.50



A Visit To PolyMorphic Systems

A National Computer Club?

Computer Conferencing

Altair vs Imsai

Software Comparison

Computer Games:
•Rotate •Nomad

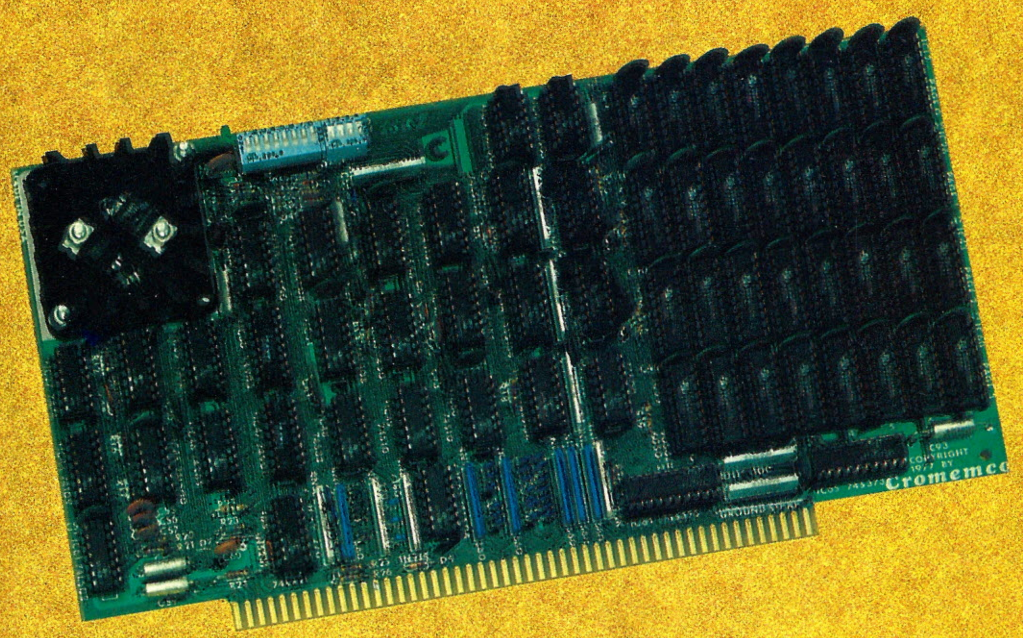
Colorful Metacolor

MINIFLOPPY
DISK SYSTEM
CT-64

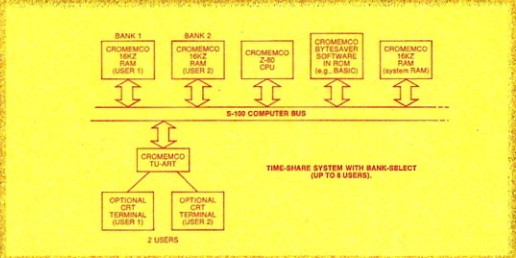
MINIFLOPPY
DISK SYSTEM

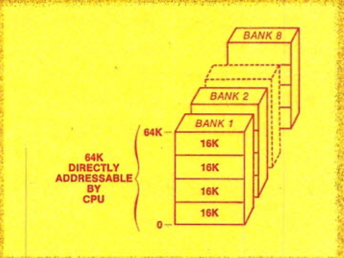
CT-64

6800 COMPUTER SYSTEM



The new 16K RAM card that turns your computer into a working giant





Available now — store/factory

Here's the industry's leading 16K RAM card.

It has two outstanding features that make it important to you:

(1) It's fast: It operates up to 4 MHz **with no wait states.** That's important because it lets you run programs on your Cromemco Z-1 and Z-2 computers in about half the time required by other systems. Even if your present computer is not 4 MHz fast, this new Model 16KZ RAM equips you for the time when you'll need and want higher computer speed.

(2) **It has Cromemco's Bank-Select feature.** Bank-Select lets you expand memory far beyond 64K bytes. Not **just beyond** 64K but **far beyond — up to 512K** bytes if you wish. Again, with Cromemco you get present outstanding performance plus obsolescence protection.

Bank-Select lets you organize memory into 8 banks of 64K each. The active bank is software-selected.

A useful giant

Whatever your S-100 bus computer — Cromemco, Altair 8800 or IMSAI 8080 — you can have enormous memory with the new Model 16KZ. You can run the large programs and files that make computers truly valuable — that take them out of the toy class and make them useful, producing units.

With Bank-Select you can even operate an S-100 bus computer as a time-share computer with up to 8 stations. A given memory bank can be accessed only by one station, so there is full confidentiality.

Advanced Cromemco engineering

Designing a 16K RAM card to operate at 4 MHz is a significant engineering accomplishment. That's why Cromemco with our strong engineering staff is the only manufacturer to offer such a card.

And notice that this advanced card is **available and ready for delivery —** at your store or from the factory.

16K RAM memory kit
(Model 16KZ-K) .... $495.
16K RAM memory assembled, tested, and burned in for 160 hours (Model 16KZ-W) .... $795.

Mastercharge and BankAmericard accepted with signed order. Show complete card number and expiration date.
California users add 6% or 6½% sales tax as applicable.

Cromemco
i n c o r p o r a t e d
Specialists in computers and peripherals
2400 CHARLESTON RD., MOUNTAIN VIEW, CA 94043 • (415) 964-7400

in this issue . . .

articles

This month's cover was produced using the Metacolor system of electronic graphics. An interview with the president of Metacolor, describing how the cover was made, appears on page 134.

fiction & foolishness

review & resources

things to do - games

departments

Sept-Oct 1977 Volume 3, Number 5 Consecutive Issue No. 17

Creative Computing magazine is published bi-monthly by Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. (Editorial office: 51 Dumont Place, Morristown, NJ 07960. Phone: (201) 540-0445.)

Domestic Subscriptions: 1-year $8, 2-year $15, 3-year $21. Subscription orders, change of address, P.O. Form 3579 to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. Call 800-631-8112 toll-free to order a subscription (to be charged only to a bank card).

Foreign Subscriptions:
Great Britain. 1-year £ 6.25, 2-year £ 11.90, 3-year £ 16.90. Orders to Creative Computing, 60 Porchester Road, Southampton S02 7JD, England.
Europe, 1-year subscription. Austria Sch 180, Belgium BF 4000, Denmark Kr65, Finland Mr40, France Fr50, Germany DM25, GreeceDr360, Holland Dfl27, Italy L8500, Norway Kr55, Portugal Esc320, Spain Ptas700. Sweden Kr45, Switzerland SF 25. Orders to Pan Atlantic Computer Systems GmbH, Frankfurter Strasse 78, D61 Darmstadt, German Fed Rep.
Other Countries 1-year $12, 2-year $23, 3-year $33 (surface postage, U.S. dollars). Orders to Creative Computing, P.O. Box 789-M, Morristown, NJ 07960.

Second class postage paid at Morristown, New Jersey and at additional mailing offices.

...notices...

ACM 1977 Annual Conference

The ACM 1977 Annual Conference will be held on October 17-19 at the Olympic Hotel in Seattle, Washington.

James Ketchel, General Chairman of ACM '77, promises a program relevant to both the practicing professional and the researcher. The technical program for ACM '77 will include sessions organized by the ACM Special Interest Groups and Committees, with additional sessions provided for papers of general interest.

Contact: Dr. James Ketchel, P.O. Box 16156, Seattle, Washington 98116. (206) 935-6776 or (206) 623-4987.

INFO/EXPO '77

Billed as "the show for the computer pro-professional," INFO/EXPO '77 will take place October 9-12, 1977, also at the Washington Hilton in Washington, DC.

For further details, contact Sharon Bennett, Data Processing Management Association, 505 Bussee Highway, Park Ridge, Illinois 60068. (312) 825-8124.

Personal Computing Expo, New York

A trade and public show, Personal Computing Expo, featuring the latest developments and equipment in hobby computers, is scheduled to be held October 28-30, 1977, at the New York Coliseum.

According to the brochure, "New York is surrounded in depth by people who work in the computer field, by computer learning centers, universities, personal computing clubs, and thousands of others whose lives are affected by computers. From this vast potential, Personal Computing Expo will draw the hard-core hobbyist, the interested student, and, because of a highly-publicized program of introductory seminars, those who are attracted and fascinated by computer but have not had exposure to the ways and means of becoming personally involved."

Over 250 exhibitors (including *Creative Computing*) are expected to show their products, and *Byte* magazine will provide speakers, as well as experts for the seminars. Leading manufacturers have been invited to explain their microcomputer systems. Seminars and lectures are free to visitors.

Our publisher, Dave Ahl, will speak on applications of microprocessors.

Contact: Ralph Ianuzzi, H.A. Bruno & Associates, Inc., 78 East 56 St., New York, New York 10022. (212) 753-4920.

Houston Personal Computing Faire

Billed as the "first exhibition in Texas dedicated to hobby computing," the Houston Personal Computing Faire will take place Sept. 17-18, 1977, at the Shamrock Hilton Hotel.

On the schedule are "exhibits by computer hobbyists of home systems, exhibits by manufacturers of the latest in microcomputing equipment, computer games arcade, computer chess tournament, door prizes, lessons for laymen, computer-generated artwork, classes for small businessmen, and workshops for hobbyists." There will also be a meeting of the newly-formed Southwest Federation of Computer Clubs.

Further information about the Faire is available from Houston Personal Computing Faire, P.O. Box 36485, Houston, TX 77036.

Compcon's First Personal Computing Evening Workshop

Compcon 77, the fifteenth IEEE Computer Society International Conference, to be held September 6-9 at the Mayflower Hotel in Washington, D.C. will feature a Sept. 8 special evening event: a hands-on session and workshop in personal computing.

Other sessions of possible interest to Creative Computing readers are on high-level languages for micros (systems languages, Basic, assembler), and microcomputer development techniques (software development, integrating hardware and software development tools).

Compcon 77 Fall, P.O. Box 639, Silver Spring, Maryland 20901. (301) 439-7007.

MicroFair International

You are invited to join WEMA's Semiconductor Industry Group at the very first large-scale exhibition of the industrial applications of microprocessors, according to the brochure on MicroFair International, due at the O'Hare Exposition Center in Chicago on October 17-21, 1977. (Actually, it's in Rosemont, "five minutes from O'Hare Airport.")

A two-day interactive tutorial/seminar on Oct 17-18 will focus on the economic considerations faced by a company when it chooses to incorporate a microprocessor into its product.

MicroFair International, c/o Golden Gate Enterprises, Inc., 1307 S. Mary Ave., Sunnyvale, California 94087. (408) 737-1100.

Western Educational Computer Conference

San Francisco is the scene of the California Educational Computing Consortium's Fall Conference, at the Jack Tar Hotel, November 17-19, 1977.

This year the Conference is being expanded and will be the Western Educational Computer Conference. Over 2000 "decision-making" administrators and faculty from over 300 educational institutions have been invited, from Arizona, Nevada, Washington, Oregon, Idaho, Montana, Utah, and California.

A very welcome innovation is the setting aside each day of certain periods when there will not be any of the scheduled 55 workshops and presentations. This time will be devoted to "the viewing of vendor exhibits."

Walter C. Wesolowski, The Claremont Colleges, Seaver Computer Center, The Claremont Colleges, Claremont, California 91711. (714) 626-8511, Ext. 3228 or (714) 982-9236.

Canadian Personal Computer Show

Described as the first computer show to be open to the public in Canada, the Personal Computer Show will be held November 4-6, 1977, at the Toronto International Center on Airport Road. The show will take place just before the eighth annual Canadian Computer Show, which will be held November 8-10 in the other two buildings. (The Canadian Computer Show is a trade show and is not open to the public.)

Contact: Derek Tidd, Canadian Computer Show, 481 University Ave., Toronto, Canada M5W 1A7. (416) 595-1811.

SIGCSE/CSA Call for Papers

Papers are invited for presentation at the Winter SIGCSE/CSA Symposium to be held at the Plaza Hotel in Detroit, Michigan, on February 23-24, 1978. This joint symposium is sponsored by the ACS Special Interest Group on Computer Science Education and the Computer Science Association, the university sub-group of the Canadian Information Processing Society. It will be held in conjunction with the 1978 ACS Computer Science Conference which will be on February 21-23 at the same location.

Papers are invited from the general area of computer science education, and are especially solicited in areas such as computer science education in developing countries, cost-effect introductory courses, team programming in courses, etc. Four copies of the complete paper must be received by November 1, 1977.

Kenneth Williams, SIGCSE/CSA Symposium Chairman, Computer Science Group, Western Michigan University, Kalamazoo, Michigan 49008. (616) 383-6151.

Mimi '77

Although Mimi 77 sounds like a girl introducing herself at a college reunion, it's actually short for the show to be held in Montreal on November 16-18, 1977, by the International Society for Mini- and Microcomputers. Mimi '77 will cover all aspects of small-computer technology, including personal and home computers.

A call for papers asks for abstracts of 200 to 250 words to be submitted by September 1 to the Symposium Chairman, Prof. J.K. Houle — Mimi '77, Ecole Polytechnique, Case Postale, Succursale A, Montreal, Quebec, Canada H3C 3A7.

Conference on Computing in the Arts and Humanities

Warren Weaver Hall at New York University is the locale of the Conference on Computing in the Arts and Humanities, Oct. 21-23, 1977, sponsored by ACM/SIGLASH (Special Interest Group on Language Analysis and Studies in the Humanities) and NYU Departments of Computer Science, Linguistics, Art and Art Education, Music and Music Education.

The program will include performances of computing in the visual arts, of music composed with the aid of computers, a computer fashion show, and a dance demonstration with computer-controlled lighting. There will be sessions in music, art and language understanding.

Contact: Dr. Naomi Sager, Conference Chairman, NYU Linguistic String Project, 251 Mercer St., New York, NY 10011. (212) 598-2294/5.

Our Face is Red

The address given in the July-August issue's "Compleat Computer Catalog" for the Pascal User's Group is incomplete. It should have read:

Pascal User's Group
c/o Andy Mickel
University Computer Center: 227 Exp Eng
University of Minnesota
Minneapolis, MN 55455

...notices...

Mini/Micro 77

The 1977 Minicomputer/Microcomputer Conference and Exposition, billed as "the largest computer show to be held in Southern California in over two and one half years" will take place December 6-8, 1977, at the Anaheim Convention Center.

Expected attendance is 10,000; the initial 1976 event drew 7,000. About 125 speakers are scheduled for the 24 presentations, "of interest to people who use or design minicomputer/microcomputer systems or peripherals. Speakers will include those from the manufacturing, financial, and educational communities ... plus more." Over 125 companies are expected to exhibit their products.

Mini/Micro Computer Exposition, 5544 E. La Palma, Anaheim, California 92807. (714) 528-2400.

Personal and Small Business Computer Expo

The Washington Hilton is the site of the Personal & Small Business Computer Expo, to be held January 13-15, 1978 in Washington, D.C.

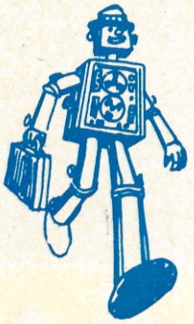
For further information, contact Felsburg Associates, Inc., P.O. Box 624, Seabrook, Maryland 20801. (301) 459-1590.

Eurocomp 78

The European Computing Congress 1978 will be held May 9-12, 1978, in London at the new Wembley Conference Center. The theme of Eurocomp 78 will be "Information Dynamics" — the control, regulation and flow of information within networks of computers. Eurocomp is said to be the first major international congress which will set out specifically to examine the design, application and implications of computer-based information systems in the light of recent developments in distributed processing, database organization, computer networking and data-protection legislation.

Papers are currently being sought (the deadline for submission of a 1,000-word resume was this last July 29) on areas such as the impact of on-line information systems, information systems design, multi-mini systems, distributed and dispersed systems, databases in an on-line environment, systems economics and feasibility, cost/benefit analyses, operations management of the information system, failsafety and security measures, national and international data-protection legislation, the cost of privacy, transfrontier data regulation, etc.

Contact: Eurocomp 78, Online, Cleveland Road, Uxbridge UB8 2DD, England, Uxbridge 39262.



State-of-the-Art vs. Compatibility

Although I have been a computer user since the IBM 650, Bendix G-15 and other assorted antique relics, I am not, by any stretch of the imagination, a Hardware Giant. Or Software Giant either. Indeed, all I want is a nice docile machine which will do my bidding and be my tool without me having to learn about its guts. In other words, I want to get in the car, turn a key, and drive off . . . . I don't want to build, repair, or even understand the durn thing.

But unfortunately it appears that the manufacturers in the microcomputer industry don't see it quite this way. First of all, most of the companies were started by engineers or programmers, exceptionally talented from a technical standpoint. Each one has a slightly better mousetrap, never mind that it's not quite compatible with one introduced three months before by another company or even their own firm. So we have a rapid succession of fantastic, state-of-the-art products, most of which don't quite work with the system you bought a couple of months ago. In fortunate cases only a few components are obsoleted; perhaps a marginal board can be updated. But remember, I'm just a user, so forget about bypassing the 7504 and putting in a 7807. Or rewiring my cassette recorder so the microphone switches off at full treble instead of full bass. Mumble, mumble. Unfortunately all too often the succession of state-of-the-art advances from the manufacturers looks like ERISA to the consumer. (ERISA to the Feds means Employee Retirement Income Security Act; to DP types it's a software package; however, as used here it means "Every Ridiculous Idea Since Adam.")

I could cite any number of real live examples revolving around the five *Creative Computing* systems but I really don't want to single out any five or six manufacturers — the problem is so very pervasive. For example, in tape cassette formats on our systems alone we have:

1. CUTS (Processor Technology SOL-20)
2. Altair (MITS. We don't worry much about this one; it hardly ever works, but we keep trying).
3. Xitan (TDL)
4. Byte/KC (SWTPC)
5. Tarbell

And why, he asked rhetorically, should software written for the Cromemco Z-80 CPU not work on the TDL Z-80 ZPU? Ha! Only the beady-eyed programmer knows. Or why, he asked cynically, should Seals and ECL 8K static memory behave entirely differently in a Dazzler-equipped system? Double ha!

Not that I should be giving advice to manufacturers and retailers, but in their position I think I would ask: are the expected sales as a result of introducing this latest state-of-the-art advance greater than the expected lost sales as a result of incompatability with existing systems?

In the short run, unfortunately, the burden is on the customer (and perhaps on the retail computer stores) to figure out if board X will work with system Y. Caveat emptor.

David H. Ahl

put...input/output...in

Poetry Anyone?

Dear Editor:

I am trying to locate some poetry about computers. If any readers know of any, I would appreciate hearing of it.

D. Van Tassel
Computer Center
University of Calif.
Santa Cruz, CA. 95064

SWARMS and EUCHRE

Dear Editor:

In your May-June 1977 issue you published two very imaginative games — SWARMS and EUCHRE. I was able to implement SWARMS on my Compucolor 8001 using color graphics for a continual display of the U.S. map and casualty report in less than 12K bytes of BASIC. The game is well-designed and plays well. The second game, EUCHRE, has an almost unbeatable strategy. I think that a very good player would not be able to beat the computer consistently because the program cheats! First, when the computer decides on trump, it looks at all 12 cards in each hand. Second, when the computer leads a card, it examines all the opponent's playable cards (your board and hand) before making its decision. These two improper strategies could be naively fixed by changing a few FOR loop indices, but this would degrade the computer's playing ability considerably because its decisions would be based on only the exposed cards and not on statistical probabilities or knowledge gained through play. While possibly this particular version of EUCHRE should be abandoned, the game itself is an ideal place for developing probablistic game strategies. The game of EUCHRE is a forerunner of Bridge and is a great deal simpler to implement than Bridge.

Gregory F. Whitten
74 Craigie St., Apt. 10
Somerville, MA 02143

A for Effort, Zero for Arab

Dear Editor:

On reading the Jan-Feb 1977 issue of *Creative Computing* more carefully, I came upon the item (p 60) titled, "A for Effort, Zero for Arab" which was listed as Anonymous. I believe the author should be given credit for his work.

The piece was written by Mr. William J. Wiswesser and first published in May 1950 under the title, "Prologue: The Empty Column, A Parable about a 'New Notation" of Long Ago."

It was rediscovered and printed again in *Computers and Automation,* January 1970 with a request for the name of the author which was provided in the March 1970 issue.

Mr. Wiswesser's address is 3103 River Road, Reading, Pa. 19605.

I believe his article came about because of his frustration in seeing a delay in acceptance of another "new" notation, which he has developed. It is a compact, linear notation for chemicals which is quite suitable for computer processing, unlike the usual two-dimensional "benzene ring" drawings taught in chemistry courses.

G. Truman Hunter
31 Overlook Drive
Greenwich, CT 06830

Corrections to Calculator Letter

Dear Editor:

I should correct two hastily-written passages in my letter you kindly published (p. 10) in the March-April 1977 issue: (1) I should have emphasized, that regarding the SR-52, I was speaking about what one infers from reading its advertising literature, not about the actual features of the device. The context is my remark about "register arithmetic." (2) Where I say that the Texas Instruments family boasts "indirect storage," the reader is entitled to conclude that this is a feature of the SR-56. It is not. Nor is it the case, as my sentence wrongly implies, that no HP devices have this feature.

My apologies. I should have amplified my original comments with the above at the time. James Blodgett, by the way, has since shown me a quite captivating scheme for addressing separate digits in a calculator's storage registers. It draws out fully the reasons for his warnings about the ten-to-the-power function where slightly inaccurate, and I am convinced he is right as rain.

George Thompson
2115 Luray Ave., Apt 2
Cincinnati, OH 45206

The CRT Syndrome

Dear Editor:

Is it true that continuous exposure to cathode rays can cause blindness, insanity, genetic defects, feeble-mindedness and death? I would enjoy hearing from any readers who have information on the subject, as I seem to be developing symptoms of all of the aforementioned maladies and naturally attribute the causes to occupational hazards.

Ken Roberts
RFD#1, Box 91
Canaan, ME 04924

Ed. Note: Two *New York Times* copy editors, who work all day at CRTs that are part of an electronic editing and typesetting system, have developed cataracts. The two editors suspect radiation from the CRTs as the cause; tests of the equipment are underway.

A Plug for PASCAL

Dear Editor:

I am writing in response to the letter by Tom Allen (p 43) in the March-April 1977 issue of your magazine. Mr. Allen's view on the inadequacies of *Basic* and Flowcharting as the standard medium for algorithms in *Creative Computing* is eminently sensible, but his proposal for a *Basic-Cobol* hybrid as such a standard is beyond belief.

There is already a language which expresses algorithms rigorously without the illegibility of *Algol,* and which has the further virtues of begin extremely easy to learn, very powerful, and already implemented. That language is *Pascal.* Consider for example how the main algorithm of Mr. Allen's "Guess a Number" program would look when written in *Pascal:*

```
printinstructions;
repeat getcomputernumber;
  repeat humanguess
  until guesscorrect
until b;
```

Could Mr. Allen really claim that his version expresses the underlying algorithm as clearly as this? Proof to the contrary is to be found in the fact that his version contains no terminating condition; this is not immediately obvious from his program, as it would be if the "b" were omitted from the *Pascal* program.

As *Pascal* interpreters and compilers become available for the 8080, the Z-80 and all the rest later this year, the use of anything but *Pascal* as the standard for personal computing programs will become less and less excusable.

David A. Mundie
104B Oakhurst Circle
Charlottesville, VA 22903

Music-Reading Machine

Dear Editor:

Susan Strapac's statement (March-April 1977, p 102) that "Michael Kassler's music-reading machine ... has not since produced any reportable results" is about as sensible as statements—well-known to logicians—that "the present king of France is bald" and that "the present king of France is not bald." No physical device has yet existed that reasonably could be described as "Michael Kassler's music-reading machine."

However, subsequent to my essay, which was written in 1963, two doctoral dissertations, by Dennis Pruslin and David Prerau, were accepted on this topic by M.I.T. A concise statement of what these authors accomplished, and how much remains to be done, can be found in my review, "Optical Character-Recognition of Printed Music: A Review of Two Dissertations," that appeared in *Perspectives of New Music,* vol. 11 no. 1 (1972), pp. 250 - 254.

Michael Kassler
Basser Dept. of Computer Science
School of Physics (Bldg. A28)
University of Sydney, N.S.W.
Australia 2006

COMPLEAT COMPUTER CATALOGUE

We welcome entries from readers for the "Compleat Computer Catalogue" on any item related, even distantly, to computers. Please include the name of the item, a brief evaluative description, price, and complete source data. If it is an item you obtained over one year ago, please check with the source to make sure it is still available at the quoted price.

Send contributions to "The Compleat Computer Catalogue," *Creative Computing*, P.O. Box 789-M, Morristown, NJ 07960.

MAGAZINES, JOURNALS

MAEDS JOURNAL OF EDUCATIONAL COMPUTING

The Minnesota Association for Educational Data Systems (MAEDS) has started publishing an annual Journal of Educational Computing. The first issue (Spring 1977) contains several articles each in the areas of instructional computing, career education, and management information systems, plus departments on applied computing reports and new publications. Subscription rate is $3 per year, or free with MAEDS membership ($10).

MAEDS, 1925 W. County Rd, B2, St. Paul, MN 55113.

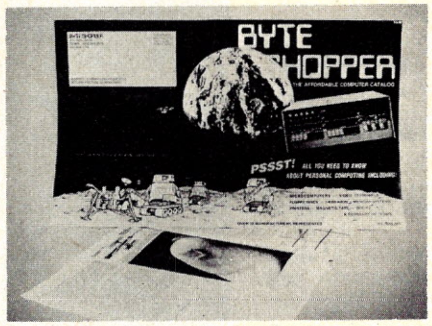
THE LOGLANIST

This is the scholarly publication of those people concerned with the constructed human language Loglan. A forum for those people trying to actually use Loglan for human and machine communication. Available through the Institute are a Loglan dictionary and grammar, cassettes, flash cards, and "pen pals." Loglan was originally developed as a tool for investigating the Sapir-Whorf hypothesis of Anthropology and Linguistics. Six issues per volume, $10; institutional rate $20.

The Journal of the Loglan Institute and all Loglanders. P.O. Box 1785, Palm Springs, CA 92262.

[Ed note: a future "Topics in Logic" article will deal with Loglan.]

VENDOR LITERATURE

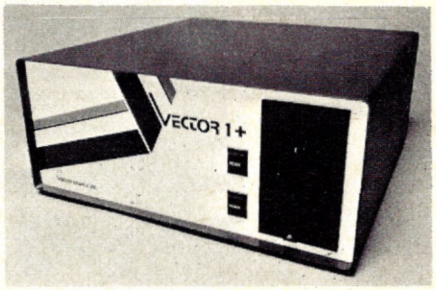


BYTE SHOPPER CATALOG

Byte Shops of Arizona has a fascinating, oversize (11″ by 14″) catalog containing an introduction to microcomputing, descriptions of various computers, peripherals, systems, manufacturers' specs, word-processing systems, and a glossary of buzzwords. Forty pages for $2.

Byte Shopper, Box 28106, Tempe, AR 85282.

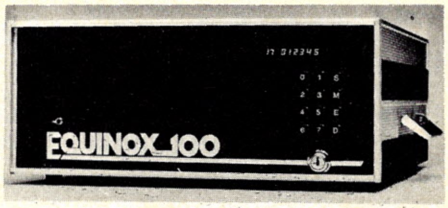
COMPUTERS



VECTOR COMPUTER WITH BUILT-IN FLOPPY DISK

The Vector 1+, for business applications, has provisions to incorporate a Shugart mini-floppy or an exact equivalent. Load and store programs within seconds. You get a cabinet in either Vector Graphic Green or Burnt Orange, an 18-slot, fully-shielded motherboard, S-100 bus, 6 connectors, a power supply 18A, 8V; 2.5A ± 16V. A whisper fan and a power-supply card to modify the Shugart drive are also included. In addition you may purchase the 8080-based CPU board with 8-level vectored interrupts and a real-time clock. A PROM/RAM board with 1K of RAM and room for 2K of PROM with a 512-byte resident monitor programmed on 2 1702A PROMS designed for an I/O board of your choice is also available. In addition, the unit requires an I/O board and a terminal or video board, keyboard and a monitor. Prices start at $389.

Vector Graphic Inc., 790 Hampshire Road A-B, Westlake Village, CA 91361, (805) 497-0733.

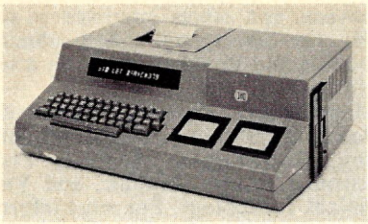


PARASITIC ENGINEERING EQUINOX 100

Offering an intelligent front panel the Equinox 100 8080-based mainframe kit is deceptively simple, with only a 12-pad keyboard and numeric 7-segment LED display. Working with the front-panel keyboard and display, the operator can monitor or alter any register, register pair, memory location or I/O device in the system. Equinox 100 can single-step through programs, slow-step at a programmable rate from 1 to 64K steps per minute, or HALT at predetermined points without "going to sleep."

The Equinox 100 also features a 20-slot busboard which is fully shielded by interlaced ground lines and cross-connected ground planes, and has active termination on every bus line. The busboard, integrated CPU/console board and optional 4K and 8K memory boards are supplied by Morrow's Micro-Stuff of Berkeley, California. The mainframe kit is under $700.

Parasitic Engineering, Box 6314, Albany, CA 94706.

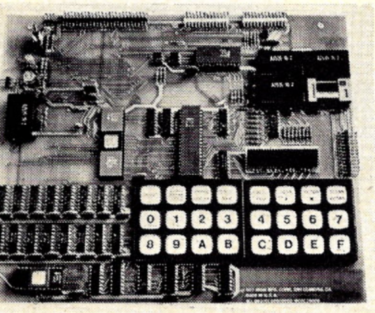


DIGITAL ELECTRONICS SELF-CONTAINED SYSTEM

The Digital Electronics Corporation DE68DT microcomputer is a self-contained system with keyboard, 20-column alphanumeric display, 40-column impact printer, single or dual mini-cassette tape drives, miniature floppy disk, RS232-C ports for supporting optional CRT, printer, or modems, numeric keypad, and nine-slot card cage with 6800 CPU and power supplies, RAM expansion to 65K bytes, and 16K-byte EPROM cards are available options which can be installed easily by the user. Software available includes assembler BASIC and FORTRAN IV, as well as a 6K-byte ROM/PROM operating system called "DEbug."

Two versions of the DE68DT are available. The standard DE68DT is a desktop model which weighs less than 30 lbs. depending on options, and is 22¼ inches wide. The DE68C is a compact unit without integral miniature floppy disk, but otherwise similar to the DT. Weight is under 25 lbs. width is 19½ inches. The microcomputers are priced from $2200.

Digital Electronics Corp., 415 Peterson St., Oakland, CA 94601. (415) 532-2920.



IMSAI SINGLE-BOARD CONTROL COMPUTER

Imsai's 8048 control computer is based on Intel's 8048 chip that contains an 8-bit microcomputer, 2.5-microsecond instruction cycle, 96 instructions, 1K words of ROM or compatible EPROM program memory, 64 words internal register memory, 27 I/O lines, internal timer/event counter, reset circuit, interrupt circuit, and a single 5-volt supply.

To create a one-board user-programmable controller suitable for use with model railroads, energy conservation systems, ham radios, household appliances, lights, etc., the system includes a cassette interface, serial I/O, five relays, 1K (optional additional 1K) of user-programmable program memory, and operation from DC or batteries. The keyboard includes 16 hex keys and 8 control keys.

With ROM monitor on the 8048 chip itself: $249 kit, $299 assembled; with monitor on 2K EPROM: $399 kit, $499 assembled. Power supply, $99.

Imsai Manufacturing Corp., 14860 Wicks Blvd., San Leandro, CA 94577. (415) 483-2093.

DECSTATION COMPUTER

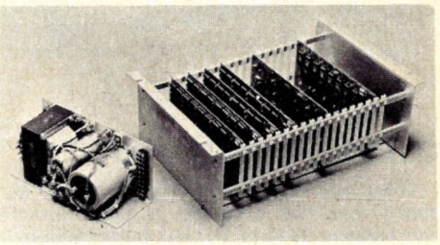
Digital Equipment's DECstation is a new-generation minicomputer system incorporating both large-scale integration and PDP-8 technology. Priced less than $8,000, DECstation is Digital's lowest-priced diskette based computer system.

DECstation employs a single-chip version of the PDP-8 central processing unit, which is integrated into a video terminal. For the system, floppy disks are used as the mass storage medium; system components are modular as "black boxes" that plug into the central system element. The central element of DECstation is a totally integrated minicomputer and display unit, the VT78. The mini omputer is an LSI version of the PDP-8 with a 16K word random-access memory. The display is a DECscope-type video terminal with both alphabetic and numeric keypads, upper and lower case ASCII character set, special symbols, and user-defined special-function keys.

Software for DECstation includes both FORTRAN IV and BASIC compilers for highlevel program development. DECstation operates under the OS/78 executive, which is resident on floppy disks. Special programs, including special-purpose "bootstrap loaders," can be read into the VT78 via a special "program injection" module that attaches to the terminal. Digital's RTS/8 real-time monitor enables DECstation to have real-time interfacing and to perform multitasking.

The standard configuration for DECstation is the VT78 video data processor and a dual floppy-disk drive, at $7,995.

Digital Equipment., Maynard, MA 01754.

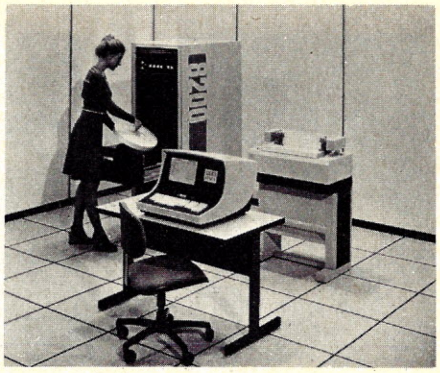


RACK-MOUNT VECTOR 1

Vector Graphic Inc. announces a rack-mount version of its Vector 1 computer. Kit includes card cage, 18-slot motherboard assembled and tested with 18 connectors, card guides and locking buttons for 18 cards. The motherboard is fully shielded to reduce noise on the bus. $225.

Heavy-duty modular power supply is also available. The 18A 8V, 2.5A + 16V custom supply provides sufficient power for full 18 boards. $90, including mounting bracket, fuse and all hardware.

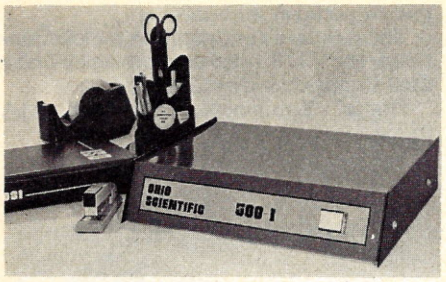
Vector Graphic, Inc., 790 Hampshire Road, Westlake Village, CA 91361. (805) 497-0733.



NCR SMALL BUSINESS COMPUTER

A small business computer designed for use as a free-standing system for the first-time computer user and as an element in a communications network, the NCR I-8230 is a conversational system capable of handling up to five separate programs concurrently. The system can accommodate all interactive industry-oriented application programs developed by NCR including manufacturing, whole/distribution, transportation, financial, health-care, government and education programs. The basic system includes a 48K processor, single cassette, CRT, 5 Mbyte Disk capacity, integrated disk controller, 55 lpm matrix printer. Rentals begin at $855 a month.

NCR Corp., Dayton, OH 45479.

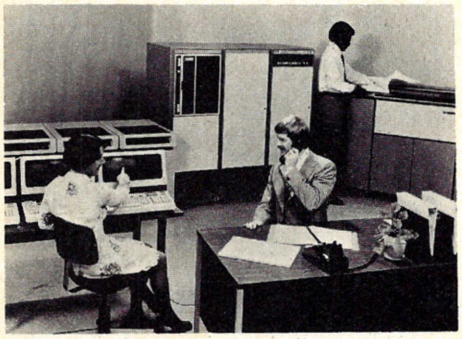


OSI BASIC COMPUTER

Ohio Scientific's new Model 500 CPU board can be used as a stand-alone computer or as the CPU in a larger system. The board accepts 8K of ROM, 4K of RAM, 750 bytes of PROM, an ACIA based serial port, a 6502 processor, and full buffering for expansion. The 500 is software and hardware compatible with Ohio Scientific's 400 kits and Challenger products, thus, allowing expansion to a large system.

The 500 is available completely assembled with 8K BASIC in ROM for only $298. By simply adding a terminal and power supply, the user has a complete system which will accept up to 200 lines of BASIC program without expansion. The board is available enclosed with power supply as the 500-1 at $429 and is available in an eight slot Challenger case as the 500-8 at $629.

Ohio Scientific, Hiram, OH 44234.



HP 3000 SERIES II

Two new models of HP 3000 Series II computer systems, with throughput and response time up to 20 percent faster than previous models, were introduced by Hewlett-Packard Company. The new Series II Model 6 and Model 8 achieve higher performance through the use of the new, faster 50-Mbyte HP 7920A disc drive. The new business computer systems were designed for use by small organizations seeking a general-purpose computer and for large organizations planning to add or expand satellite EDP capability. Educational institutions are said to have found the Series II to be cost-effective in handling both their administrative and their computer instruction applications.

The Model 6, with a base price of $110,000, has as standard hardware 128 Kbytes of memory (expandable to 256 Kbytes), 50-Mbyte disc, 1600-bpi tape drive, 2640B CRT console, and a terminal controller which can handle up to 15 asynchronous terminals at speeds up to 2400 baud. Standard software includes the MPE II operating system, implementing batch operation, transaction processing from terminals, and interactive program development. The basic software package also includes SPL (a high-level Systems Programming Language), Edit, Debug/Trace, Fcopy, Sort and compiler library. The Model 8, which has a base price of $140,000, has 320 Kbytes of memory which can be expanded to 512 Kbytes. It offers the same system configuration and standard software as the Model 6.

Inquiries Manager, Hewlett-Packard Co., 1501 Page Mill Road, Palo Alto, CA 94304.

ORGANIZATIONS

CONDUIT

CONDUIT, an organization which has been working to improve instructional computing at the college level since 1972, has been re-funded by the National Science Foundation to continue its activities for at least another two years. CONDUIT distributes quality instructional computing materials for higher education.

To perform this function, CONDUIT searches for quality computer-based instruction, peer-reviews such units to validate substantive content, tests software to ensure technical correctness and reliability, packages curriculum units so that both the software and teaching techniques are transferrable, and distributes these products on a national basis.

CONDUIT now distributes 41 curriculum units covering topics in Biology, Chemistry, Math, Management Sciences, Physics and Social Sciences. More units are being actively sought; the addition of approximately thirty packages per year is planned.

CONDUIT also developes guidelines and reports. One such product will be a second set of discipline-oriented "state of the art" reports, telling instructors how they may become involved in instructional computing and what is going on currently with computers in their own discipline.

For more information (including a complimentary copy of the CONDUIT newsletter, the Pipeline) write to CONDUIT, Box 388, Iowa City, Iowa 52240.

CENTRAL STANDARDS LIBRARY

To help solve some of the standards problems in the hobbyist computer and microcomputer field, ALF Products is sponsoring a Central Standards Library as a means of standards information exchange for manufacturers, consumers, hobbyists, and others interested in standards. The Library will collect submitted standards and distribute them on a non-profit basis.

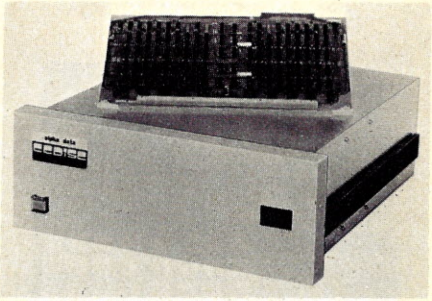
Manufacturers currently participating include: ALF Products, IMSAI Manufacturing, PolyMorphic Systems, Proko Electronics, Vector Graphic, and Video Terminal Technology. For more information on available standards, on how to submit standards, and on the library's services, send $1 (to cover printing and mailing costs) to: The Central Standards Library; c/o ALF Products Inc.; 128 S. Taft; Denver, CO 80228. You will receive a copy of the first CSL Newsletter and the first submitted standard (a parallel interface standard).

NATIONAL COMPUTER CLUB

Organized to provide communication among hobby-computer clubs, the National Computer Club has been formed. Over 45 club leaders and representatives met at the first Club Congress in June at NCC in Dallas. A committee was formed to discuss the services that should be provided by a national society for personal computing, and will meet at various computer shows.

The Committee's secretary is Jim White, 1202 River View Lane, Watertown, WI 53094.

PERIPHERALS



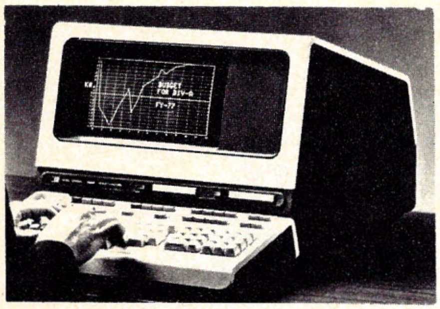
SOLID-STATE DISC MEMORY

The solid-state charge-coupled-device Semiconductor CCDISC memory, by Alpha Data is a disc that doesn't rotate. It is a disc without heads, platters, bearings or motor; it has no moving parts. The CCDISC has an ultra-fast average access time of 250 microseconds, almost 40 times faster than the fastest electromechanical rotating disc memory. CCDISC memories are said to be ideal for fast buffer, paging, scratch-pad, program-swapping, time-sharing, and distributed network applications.

The CCDISC has a capacity of up to 1024K bytes in 128K-byte increment plug-in PC boards. Capacities of up to 4 megabytes can be achieved by daisy-chaining. $3,195 in OEM quantities.

Alpha Data Inc., 20750 Marilla St., Chatsworth, CA 91311, (213) 882-6500.

TERMINALS



HP GRAPHICS TERMINAL

Hewlett-Packard's HP 2648A is a new microprocessor-controlled graphics CRT terminal, using raster-scan technology, features a bright display, selective erase, independent graphics and alphanumeric memories, automatic plotting of tabular data, rubber-band line, system-independent zooming and planning. Although specifically designed for graphics applications, the 2648A also has the data entry and data communications capabilities of the proven HP 2645A display station.

Zooming and panning the image is a major feature of the terminal and can be done with a single keystroke without support of a cpu. Any portion of the graphics memory can be magnified up to 16 times to permit investigation, modification and redrawing of especially dense areas. Concurrently, users may pan any portion of the magnified display that is not in the viewing window. Panning through the expanded display does not require reinitialization of the display data.

The terminal offers a rubber-band-line capability which allows users to stretch and then draw a line to any length in any direction between a selected point and the cursor. The rubber-band-line speeds graphics development by enabling users to draw trial sketches, such as architectural floor plans, with or without connecting the terminal to a CPU. To improve architectural and mechanical-part graphics, the terminal offers area shading and pattern generation.

Base price for the Hewlett-Packard 2648A graphics display terminal is $5,500; when equipped with cartridge tape drives, the price is $7,100.

Inquiries Manager, Hewlett-Packard Co., 1501 Page Mill Road, Palo Alto, CA 94304.

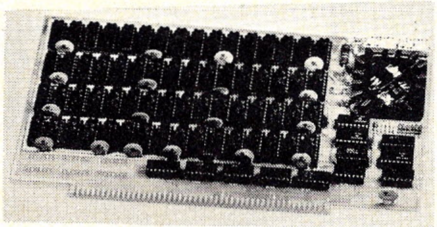
MISC. HARDWARE

KIM-1 ADD-ONS

The Computerist has several products for use with the KIM-1. A 4K RAM board with 2102 static RAM chips is 5¼" by 9¼", fits over the top half of a KIM-1; $129. assembled. The HELP editor package, which performs text and source editing, is $15 on cassette. The HELP mailing-list package will create, maintain and print mailing labels and lists; it puts about 900 names on one side of a 30-minute cassette; $15. A power supply for the KIM-1, assembled and tested, is $40.

A digital-to-analog converter provides a complete audio input system, includes a cassette tape with a four-part-harmony music program; $35.

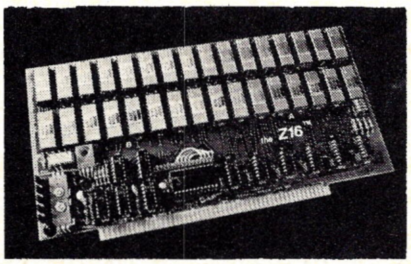
The Computerist, Box 3, S. Chelmsford, MA 01824. (617) 256-3649.



LOW-POWER MEMORY BOARD

The Logos I, from Advance Microcomputer Products, is an 8K fully static low-power memory board for the Altair S-100 bus. It features DIP-switch selectable addressing on any 1K boundary and a hardware memory-protect circuit that allows protection of one 8k memory block, two 4K blocks, four 2K blocks, etc., up to thirty-two 256-byte blocks. Logos I runs at full spead with no wait states. On-board battery backup is included. Kit includes low-profile sockets for all ICs, at $248; assembled; introductory price $219.95 kit.

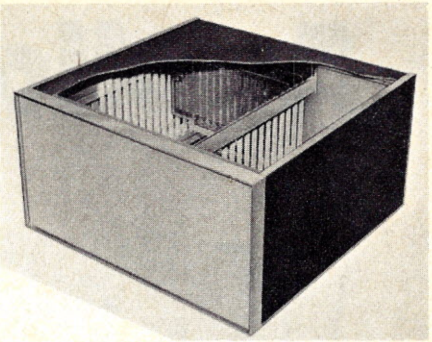
Advanced Microcomputer Products, Inc., P.O. Box 17329, Irvin, CA 92713. (714) 558-8813.



TDL 16 K RAMBOARD

Technical Design Labs announces a 16K static RAM board, the Z16, for the Altair S-100 bus, featuring extremely fast access times, very low power consumption, and low cost. Optimization of on-board logic gives the Z16 a worst-case access time of 250 nsec, thus it can run without wait states at up to 4 MHz or more. Any 4K block may be individually addressed at any 4K page border, and each 4K block may also be individually switch-protected. $574 kit, $699 assembled.

Technical Design Labs Inc., Research Park, Bldg. H, 1101 State Road, Princeton, NJ 08540. (609) 921-0321.

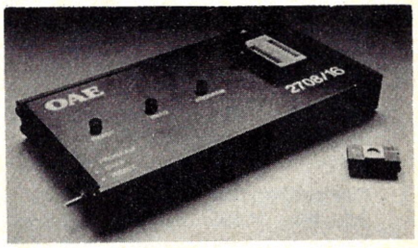


VECTOR-PAK ENCLOSURES

For system expansion or new construction, two new low-cost enclosures, from Vector Electronic Company, give system developers an alternate choice in packaging IMSAI, Altair, and other 5.31-inch by 10-inch (13.48-cm by 25.4-cm) cards with S-100 bus configurations. The cases are form-compatible with IMSAI and Altair microcomputers, the VP2 having front-to-back card orientation while the VP1 has side-to-side card orientation. A sturdy aluminum chassis at the rear or side supports power supplies and other heavy components.

Two dozen plastic guides are supplied separately so that the user may position 12 cards in any location and with any board spacing. For expansion, the cases have space for 21 cards on 0.75-inch (1.90-cm) centers. Adjustable slots allow mounting of receptacles or a motherboard. The VP1 is $128.30; the VP2, $134.30.

Vector Electronic Company, 12460 Gladstone Avenue, Sylmar, CA 91342, (213) 365-9661.

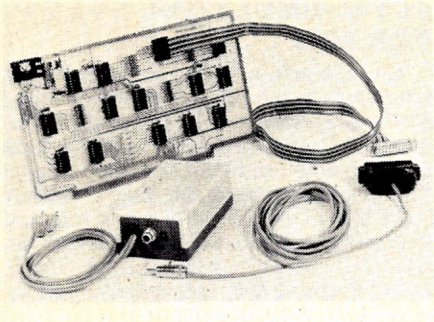


PROM PROGRAMMER

OAE's PROM programmer the PP2708/16, programs the industry-standard 1K 2708 and the new 2K 2716 PROMs from Texas Instruments. A simple parallel interface connects the PP 2708/16 to any microcomputer. An internal address counter simplifies interfacing. Only one unregulated 8V supply is required. Very little software is required to support the programmer. Simply dump the data via an output port to program a PROM.

The programmer contains address counters, timing and control logic, and a DC-to-DC regulated power supply. Each unit comes complete with a black anodized aluminum case, a five-foot ribbon cable with pre-wired connectors, and software. $249, kit; $299, assembled and tested.

Oliver Audio Engineering, Inc., 7330 Laurel Cyn. Blvd., N. Hollywood, CA 91605. (213) 765-8080.



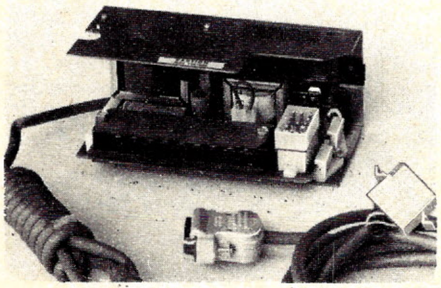
POWER CONTROL SYSTEM

Comptek's PC3200 Power Control System, compatible with the Altair S-100 bus, is an AC power-switching system that enables microcomputer control of lights, small motors, appliances, tools, etc. To eliminate lengthy runs of AC power cabling, the system offers Control Logic Interfaces on computer boards, and Power Control Units that can be remotely located at the point of control. Control outputs from the interfaces are low-voltage, current-limited signals, with optical isolation at both ends of the control signal for maximum noise immunity and short-circuit protection.

The interfaces contain up to 32 independently addressable control channels, but require only one output-port address. A single-byte output from the processor selects an individual channnel, and turns it on or off without affecting the state of any other channels.

The 16-channel Control Logic Interface is $189 kit, $240 assembled. The 400-watt, 120-VAC Power Control Unit is $39.50 kit, $52 assembled.

Comptek, P.O. Box 516, La Canada, CA 91011.



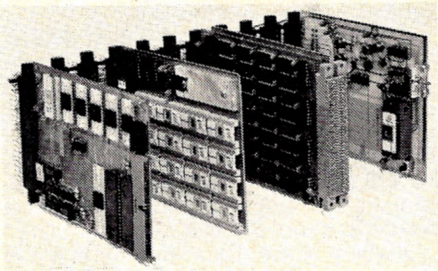
TTY/EIA/CRT/20MA ETC. CONVERTERS

United Data Services offers a complete line of converters, couplers and controls for going from Modem A to Terminal B to Minicomputer C and so on. Most of them are priced at a bargain 30% to 50% of the comparable unit from a terminal or computer manufacturer.

One of the most popular is bound to be Model 312A 0561 which converts from EIA (RS232) to 20-ma current-loop TTY terminals. Cost is only $45.00. I recently got a standard 20-ma TTY which I wanted to alternate with a CRT terminal with my Altair—this United Data Services coupler was just the answer.

Other goodies include TTY Idle Line Motor Control units ($62.50), TTY to Minicomputer (9 types), Adapters ($115), Peripheral Equipment Adapters ($135), and many more.

United Data Services Co., Inc.; 3024 North 33rd Drive, 103; Phoenix, AZ 85017 (602) 269-2449.



WINTEK REDUCES PRICE

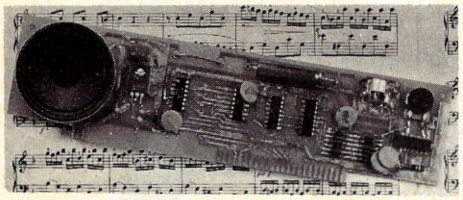
Wintek has reduced by 27% the price on their 16K byte RAM module, from $899 to $699. This follows their 50% price reduction in January on their one-card microprocessor, to $149. The Wince Micro Modules are said to be unique in that they are "the only 6800-based modules available on industry-standard 4½" x 6½" printed-circuit cards." Other Wince modules include ROM, EROM programmer, relay driver/sensor, console, and interface.

Wintex Corp., 902 N. 9th St., Lafayette, IN 47904. (317) 742-6802.

PROM PROGRAMMER

The Prammer, from Xybek, is 1702A PROM programmer, compatible with the Altair S-100 bus. The Prammer can hold up to 1792 bytes of 1702A EPROM, and also has 256 bytes of RAM. The EPROM programmer will program a complete 1702A in 18 seconds. Kit price for the Prammer is $189. A 3-foot extension cord kit at $15 permits any of the seven 1702A sockets to be extended to a zero-insertion-force socket outside the computer, for programming and copying quantities of 1702A's.

Xybec, P.O. Box 1631, Cupertino, CA 95014. (408) 296-8188.



MUSIC BOARD

Newtech Computer Systems' Model 6 Music Board enables anyone with an S-100 bus computer to produce music and sound effects. Applications include generating melodies, rhythms, sound effects, Morse code, touch-tone synthesis, and much more.

The model 6 comes fully assembled and tested, and features include selectable output port address decoding, a latched 6-bit digital-to-analog converter, audio amplifier, speaker, volume control and RCA phono jack for connection to an audio system. A complete user's manual is supplied and includes a BASIC program for writing musical scores and an 8080 assembly-language routine for playing them. The Model 6 Music Board is $59.95 through computer stores.

Newtech Computer Systems, Inc., 131 Joralemon St., Brooklyn, NY 11201. (212) 625-6220.

SOFTWARE

BUSINESS BASIC

Data General's new multi-user Business BASIC language package has extensions for multi-key indexed sequential file structures, file maintenance and file security, and double-precision integer arithmetic. Business BASIC is specifically designed for business data processing on Data General Nova and Eclipse computers running the Real-time Disc Operating System (RDOS). An optional interface to the INFOS data management system gives BASIC users access to INFOS ISAM and DBAM files on commercial Eclipse C/330 systems.

The addition of Business BASIC gives users a business-oriented language capability across the entire Data General processor product line, and allows programs developed for small NOVA and ECLIPSE S/130-based business systems to be transported to larger ECLIPSE systems without modification.

A typical system using Business BASIC would consist of an Eclipse S/130 computer with 128K bytes of memory, 20 megabytes of disc storage, Dasher printer console, and four Dasher display model 6053 terminals, diskette, 300 lpm printer, RDOS, Business BASIC, installation, training and support services. This system would be about $75,000.

Barbara Nolan, Data General, Route 9, Westboro, MA 01581. (617) 366-8911.

TINY BASIC

For those microcomputer users who prefer to use a higher-level language, Tiny BASIC, a subset of Darmouth BASIC, permits immediate entry and execution of Tiny BASIC language programs. ROM software has been designed so that most any I/O devices can be used.

The RAP/Tiny BASIC ROM package (SW101) is priced at $200 and includes full documentation. RAP is also available on a set of seven 1702A PROMs (SW200) for $295. Tiny BASIC is available either in paper tape format (SW300) for $25 or on a set of nine 1702A PROMs (SW201) for $275..

Contact Darrell Crow, Microcomputer Associates, 2589 Scott Blvd., Santa Clara, CA 95050. (408) 247-8940.

SMALL OPERATING SYSTEM

Are you tired of toggling in a bootstrap every time you turn your computer on? Or would you like to have more than a simple bootstrap in ROM? Then you may be interested in SOS, a Small Operating System offered by LSM Engineering, designed to be stored in ROM. SOS has commands to save, load, and verify data saved on Tarbell cassette; read both Intel hex format and binary papertapes; fill, move, and verify blocks of memory, and to enter and dump memory directly. It is supplied both as a source listing and object code (assembled at B800 hex). SOS requires less than 2K ROM, so there is room for other programs in a 2K ROM board.

If you want to do some text editing with your computer, LSM Engineering also sells a text editor, called EDIT, which can be used to modify plain text or source programs. EDIT commands permit additions, deletions, and replacement of text by character, string, line, and page. Commands may be concatenated to form command strings or be defined in mccros. Any type of terminal can be used with EDIT. Tarbell cassette-interface handlers are provided with EDIT, and complete specifications are given for writing custom peripheral handlers. SOS is available on Tarbell block format cassette for $15.00, and EDIT is available on either Tarbell cassette or hex papertape for $22.50.

LSM Engineering, P.O. Box 3243, Orange, CA 92665.

PDP-8 SIMULATOR FOR 8080

The Simul8tor is a complete PDP-8 simulator for the 8080 (or Z-80), enabling 8080 owners with at least 12K to utilize the many of PDP-8 programs available both commercially and through the Digital Equipment Corporation User's Society (DECUS). DECUS software is inexpensive and readily available, and includes ALGOL, BASIC, FOCAL, SNOBOL, FORTRAN, LISP, assemblers, editors, debuggers, floating point.

Simul8tor is available in two formats: INTEL papertape and INTEL Tarbell Cassette. Simul8tor comes complete with a User's Manual, PDP-8 Programming Tutorial, PDP-8 Loader, DECUS library information, and a source listing of Simul8tor's IO routines for users who wish to modify them. $20. Add $3.00 for each cassette ordered.

Registered Simul8tor owners will also receive discounts on Simul8tor products to come: Extended Memory options, powerful debug package, Extended Arithmetic option, and comprehensive device support.

Amide Corp., Box 600, Sag Harbor, NY 11963. (516) 725-2880.

MICROCHESS FOR THE 8080

Micro-Ware Limited announces the availability of MICROCHESS for the 8080. MICROCHESS is an intelligent chess-playing program which will run on any 8080 or Z-80 system with 4K of contiguous memory and an ASCII input/output device. Documentation includes a Player's Manual with a description of the program, and a comprehensive appendix with details for conversion of the I/O routines, or customization of the board display programs. The program is furnished on paper tape. Features of MICROCHESS include automatic board display, multiple levels of playing skill, castling, and the ability to reverse the board at any time, making it possible for the computer to play a game against itself. $15. MICROCHESS for the KIM-I is available for $10.

Micro-Ware Limited, 27 Firstbrooke Rd., Toronto, Ont., Canada. M4E 2L2.

MINI WORD-PROCESSING SYSTEM

The Software Store has announced a Mini Word-Processing System running on MITS Altair equipment under Disk Extended Basic for $150. Mini word Processing is designed to help an operator generate letters, text and mailing labels or envelopes. The system consist of seven programs which are driven by a menu-select routine from which any of the seven processing programs can be utilized. Each program interacts with the operator to establish file names and drive numbers. The options are selected by the operator using simple Y or N (Yes or No) responses to the detailed program prompts.

The Software Store, 706 Chippewa Square, Marquette, Michigan, 49855. (906) 228-7622.

OSI 6502 8K BASIC

OSI's new 8K BASIC for the 6502 was written by Microsoft, the people who wrote Altair 8K BASIC. The OSI 8K BASIC is identical to the Altair 8K BASIC, except that the OSI BASIC has automatic string space handling, and it is said to run faster. Available on paper tape, audio cassette or floppy disk, at $50. Free with the purchase of any 12K or larger OSI Challenger.

Ohio Scientific Instruments, 11679 Hayden St., HIram, OH 44234.

MULTI-USER OPERATING SYSTEM

A multi-user, multi-tasking operating system for MITS Altair 8800 computers with MITS floppy discs called TEMPOS has been announced by Administrative Systems, Inc. According to A.S.I., TEMPOS permits up to seven users to access the system concurrently running either shared or separate tasks. Output from each task is spooled to a disc file. Each task is allocated 16 milliseconds of processor time except for those which are awaiting input from a terminal. Additionally, after initiating a task a user may detach the terminal from the task and begin another.

TEMPOS supports a high-level language called OPUS/TWO. This language has extensive file-handling features and permits shared access to data files. A text editor, assembler, package of utilities, and a Clinical Accounts Receivable/Billing System are also available to run under TEMPOS. A minimum of 48K of memory is recommended for a system with three users and two discs, since all tasks are memory resident. TEMPOS is $1000; OPUS/TWO and other software packages are extra.

A.S.I., 222 Milwaukee, Suite 102, Denver, CO 80206.

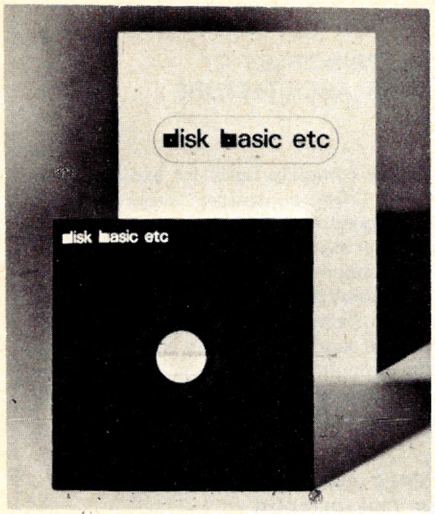
ASSEMBLER EXTENSIONS

Two software packages of interest to owners of 8080-based microcomputers have been developed by Objective Design Inc. The first of these is an addition to Processor Technology Software #1, a widely used assembler. Objective Design's Software Package 0.5 adds the features of automatic line-numbering, formatted output, text editing (insert, delete, replace) on the current line or the first occurrence of a string in the text file, and renumbering of the file. It also extends the assembler itself to include a global symbol table, output of the symbol table, octal constants, assembly of multiple sections of source code from a mass storage device, and a 'ASC' pseudo op for assembly of ASCII characters in a program. To use SPKG.5 you must already have Processor Technology Software #1 running on your system. SPKG.5 will add 1.75K to the memory requirements of P.T SW #1. In order to make the additions, one must load SPKG.5 in source code, assemble it, and then make patches in the P.T. SW #1 with which it was assembled.

Objective Design also markets a video game called "Encounter," which is a computerized board game. It is played by two people with two keyboards and a memory-mapped video display such as Processor Technology's VDM-1. Moves are made as fast as they can be typed in. Encounter can easily be reconfigured, so it is actually several different games in one. It's a bit difficult to describe Encounter, but it is a war game, a little like Chess (with some significant differences.)

SPKG.5 in source form is $14.95. Encounter (source code, game instructions and rules) is available for $12.95. It is also available on papertape for $16.95.

Objective Design Inc., P.O. Box 20456, Tallahassee, FLA 32304.



DISK BASIC

Binary Systems Corporation has introduced DISK BASIC ETC, a disk-accessing, extended version of BASIC ETC. DISK BASIC ETC, an interpreter for 8080-based microcomputers, is a general purpose program for business and scientific applications, as well as hobbyist game programming. The sector-based DOS, which works with the iCOM floppy disk controller, makes available up to three memory buffer files to the user.

The disk software includes six file manipulation commands plus SAVE, LOAD and two special integer functions helpful in keeping track of files. Extensions to BASIC ETC include new options to extant commands; for example, with DISK BASIC ETC either a line number or exacutable statement may follow THEN or ELSE; the addition of ON-GOTO and EDIT statements; the inclusion of 11 more functions; and, refinements such as increasing the execution speed of some statements. DISK BASIC ETC uses the lower 12 KB of memory plus 1 KB of scratchpad.

DISK BASIC ETC is supplied on a certified, 5 ¼-inch minifloppy disk, or on a certified, 8-inch regular floppy, along with a user's manual. The price is $50.00; the manual sells for $10.00 separately.

Micro Store, 634 S. Central Expressway, Richardson, TX 75080. (The Micro Store is the retail affiliate of Richardson-based Binary Systems, Inc.)

BASIC INTERPRETER

An *interpreter,* which translates and executes the user's program directly, featuring ease of program development, straight-forward, one-step program execution, memory-efficiency, full string capability with up to 255-character string variables, N-dimensional arrays, 27 error codes, subroutine nesting, 31 commands and statements, 8 functions plus user-defined functions, formatted output statements and variable-precision arithmetic, and resides in only 8KB of memory, is available in cassette tape (Kansas City or Suding/Digital Group format) or on paper tape.

BASIC ETC comes with a 32-page manual and was developed by John Arnold and Dick Whipple, authors of Tiny BASIC. $25.

The Micro Store, 634 S. Central Expressway, Richardson, TX 75080.

MISCELLANEOUS

LASER MUSIC SPECTACLE

A Soleil Laser Music Spectacle is made up of three types of laser movement. For the pre-programmed type, every movement of the laser light is planned, and a large-scale digital computer is used to compute the millions of commands required for a Soleil show. The program is stored on four-channel audio tape, with two channels for stereo music, and two for control information for the lasers. The laser-light movements can also be controlled directly by an operator; these movements can also be combined with a pre-programmed tape.

The Soleil Laser Music Spectacle is available for engagements. Contact Kim Rogers, RR #2, Box 103, Bloomington, IN 47401. (812) 336-8222.

CODING FORM

Walton Electronics has a new coding form for those who work in assembly or machine language. The forms bound in pads of 50 sheets, are formatted to accept code for any of the more popular microprocessors. Columns include: address, code, label, instruction, and notes. Code can be written either in octal or hex with up to three bytes per line in regular assembler format. $1.95 postpaid.

Walton Electronics, Box 503, Bethany, OK 73008.

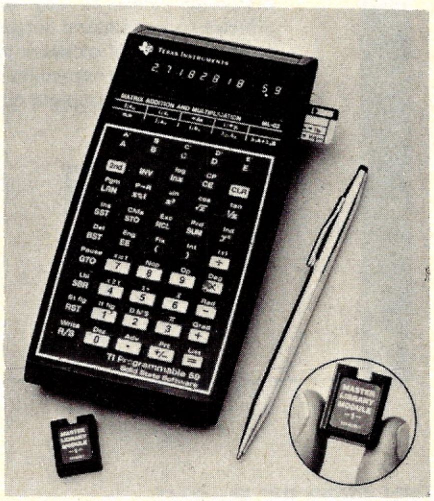
CALCULATORS

HEXADECIMAL CALCULATOR

The TI Programmer, a new handheld calculator from Texas Instruments, does arithmetic in three different number bases, and converts to and from these bases: hexadecimal, octal, and decimal.

Among other applications, the TI Programmer will convert memory address to decimal form, add relative address to a base address to find specific computer memory locations, or determine if there is enough space in the computer's memory to hold a new block of data. The calculator can also perform bit-by-bit logic operations on numbers in hex or octal, including AND, OR, Exclusive OR and SHIFT operations. The calculator is being test-marketed initially on a direct-mail basis from Texas Instruments, at $49.95.

Texas Instruments Inc., Inquiry Answering Service, P.O. Box 5012, M/A 84 (Attn: TIP), Dallas, TX 75222.



PROGRAMMABLE CALCULATORS WITH PLUG-IN SOFTWARE

Featuring plug-in, Solid State Software libraries, the new TI Programmable 58 and TI Programmable 59 calculators provide additional program memory capability and programming flexibility to advanced students and professionals in business, engineering and science. Both models can use interchangeable Solid State Software modules with prerecorded program libraries containing up to 5,000 steps each. The libraries range from applied statistics and surveying to real estate/investment, aviation and marine navigation.

The TI Programmable 59 also has magnetic card memory capability. Up to two cards about the size of a stick of chewing gum with an additional 960 program steps can be loaded into this calculator. Otherwise, the two calculators differ only in storage capacity. Users can partition the TI 58 with up to 480 program steps or up to 60 memory registers. They can do the same with the TI 59 with up to 960 steps or up to 100 memory registers. Suggested retail price for the TI 58 is $124.95 and for the TI 59, $299.95.

Texas Instruments Inc., Box 5012, Dallas, TX 75222.

Here's everything you need to program your own '8080'!

SCELBI does it again! Here, for the first time, are all the tools you need to get the most out of your '8080'. About the only thing missing is the '8080' itself. Just look at what's in this total package . . .

'8080' Octal Code Card and/or **'8080' Hexadecimal Code Card,** instant slide rule aids for programming or debugging '8080' software. Contains all standard mnemonics, with corresponding codes for either octal or hexadecimal modes. Easy-reading instructions are color coded to indicate which flags are affected during instruction execution. Front of card features quick, logical reference formats, while back displays ASCII code chart for all 128 characters, plus '8080' status words and register pair codes.

The '8080' Programmer's Pocket Guide is a compact 3 x 4½" ever-ready, instant reference for either Code Card, that explains the instruction set in detail, giving both octal and hexadecimal codes.

SPECIAL MONEY SAVING INTRODUCTORY OFFER!

Each of these "must" programming tools sells individually for only $2.95. If you order today, you can own any two for only $5.50 . . . or all three for only $7.95! You need these low cost, highly productive tools. So, order today, and save, Save, SAVE! Clip this handy mail order coupon and mail it today!

Scelbi Computer Consulting, Inc.
P.O. Box 133 PP STN — DEPT. C
Milford, CT 06460

Please rush items checked below (Write quantities of each in ◯)

◯ **Octal Code Card(s)** ◯ **Hexademical Code Card(s)**

◯ **The '8080' Programmer's Pocket Guide(s)**

- ☐ I enclose $2.95 for Item indicated above.
- ☐ I enclose $5.50 for 2 items indicated above.
- ☐ I enclose $7.95 for 3 items indicated above.

LIMIT: 3 items per customer for this money-saving Introductory Offer.

Name ____________

Address ____________

City ____________ State ______ Zip ______

A national computer club? Will it benefit the hobbyist? What could it, should it, undertake to accomplish? The chairperson at the First National Computer Club Congress at Dallas gives an insider's accounting on what may become an important organization.

It's Time For A National Computer Club

Rich Kuzmack

The first National Computer Club Congress was held in Dallas this past June in conjunction with the 1977 National Computer Conference (NCC). About thirty computer clubs across the country were represented by over forty-five club leaders and delegates, who assembled to exchange ideas and discuss issues related to club activities and programs, with great attention focused on the question of a national organization.

Club leaders have been meeting at computerfests for informal discussions since the first fest in Trenton in the spring of 1976, but the prestige of the NCC and the support of Dr. Portia Isaacson as chairperson of the 1977 NCC provided both a tremendous opportunity and a serious challenge to take action toward forming a national organization. Rather than leave a worthwhile outcome to chance, club leaders held pre-NCC planning sessions last spring at the computerfests in San Francisco, Trenton, and Cleveland. In addition, a survey of club leaders was conducted by mail to solicit opinions, especially of those unable to make it to any of the planning sessions or to the Congress itself.

The Congress had two sessions on successive days. The first session addressed the purposes a national organization might serve and the organizational structures that would be appropriate to achieve those purposes. It began with a report on the results of the survey and included a panel discussion on the organizational structures used in the regional organizations of computer clubs: Southern California (SCCS), Midwest Alliance (MACC), and Chesapeake (CMC). (A new regional group was formed at the Congress, the Southwest Federation of Computer Clubs.) A solid concensus of the Congress favoring the establishment of a national organization developed, but no firm concept emerged on the essential characteristics and purposes such an organization should possess.

By the second session of the Congress it was quite apparent that a lot more work would be needed before an organization could be established, but also, that it was time to get started. Accordingly, after considering several proposals it was decided that a working

group to be known simply as the Committee would take on the job "to define and establish a national amateur computer society." Membership on the Committee is open to anyone seriously interested in working to achieve its objectives.

The Committee met immediately following the Congress, selected a Secretary to consolidate and distribute communications among its members, and received pledges of financial support from the established regional organizations to cover the Secretary's

Photos by Tom Woodward of SCCS Interface

Big talk in Texas: the Committee meets in Dallas after the first National Computer Club Congress

Unlike other organizations that are formed from the top down, local amateur computing clubs existed in the hundreds before thought was given to the need for going national. Pictured here are three club meetings.

At the Chesapeake Microcomputer Club, manufacturers are often invited to discuss their products.

expenses. The Secretary is Jim White (1202 River View Lane, Watertown, WI 53094) of the Durant Computer Club. Each member of the Committee will be setting down ideas, for distribution through Jim to other Committee members, and will be seeking inputs from members of their respective clubs. In addition, articles will be prepared for the various publications that serve the amateur computing community to keep everyone informed and encourage participation in the effort. Finally, members of the Committee will take advantage of computerfests to meet, under a rotating presiding officer, in open forums to which all interested individuals are invited.

Both the Congress and the Committee meeting were able to operate by concensus, essentially deferring those topics on which there was not a broad base of agreement. These issues do have to be addressed and at least tentatively resolved if the Committee is going to make progress toward its objective. In discussing some of these issues I will be suggesting solutions which reflect my own personal opinion. All concerned hobbyist views are needed, and they are encouraged to write them down in a letter to the Committee's Secretary and participate in the formation of a truly representative hobbyist society.

Amateur Defined

"... to define and establish a national *amateur* computer society" is the way the Committee charter reads, but there has been some concern expressed about the term "amateur" by professionals in various aspects of the computer industry. Computer professionals were in the vanguard as the movement to personally-owned computers grew, but it is rare for any one individual to be professionally involved in the breadth of hardware and software, electronic and mechanical devices, and systems and applications programming, with which the amateur computer enthusiast must be concerned. By and large, amateur computing is not unprofessional, but at the same time it should be non-professional. The distinction between amateur and professional, it seems to me, should be based on whether or not the *computer aspects* of an activity or project are being done *for pay or commercial gain.*

Many amateurs are finding interesting work-related applications for their computers. This includes those into computers professionally who are using their own computers for professional development in their field. If I, as an economist, use my own computer to do some statistical analyses for work, that should still be considered amateur computing. A professional systems programmer who writes a routine to drive a video display for his own computer is also doing amateur computing. But the work that the systems programmer does for his employer or his clients should be considered professional and should meet professional standards for quality. And should I offer my computer skills or products for sale, then for that project I would not be an amateur and should be expected to meet whatever professional standards the marketplace imposes.

Questions of Purpose

The next step toward defining a national amateur computing society is

President Lou O'Block tries to silence heckling by John Coklet at meeting of Cleveland Digital Group.

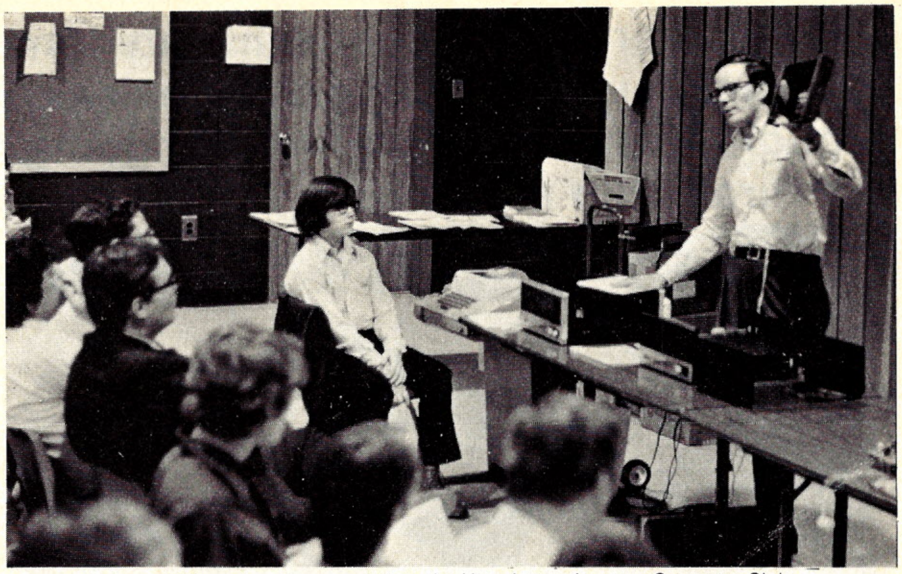
deciding on the purposes the society should serve. The basic purpose, of course, is to serve the amateur computing community, which simply raises the obvious question, "How?" Included among the many good and sufficient purposes appropriate to a national society are standards development and dissemination, computerfest schedule coordination, hardware and software products evaluation, educational and club program services, ombudsman services, and more. There are also some potential purposes that stir up considerable controversy, such as membership directories, group purchasing, and lobbyist activities.

What's the fuss about these last few? Many people at the Congress thought they were pretty good ideas, and taken at face value they seemed worthwhile. The discussion, however, brought out some good reasons against doing them. A membership directory, for example, could really help in getting people together; unfortunately, it would also serve as a handy guide for thieves to some pretty valuable loot. Group purchasing is tempting, indeed, until all the benefits, costs, and alternatives are considered, and then it doesn't fare too well. It would be an administrative nightmare unless run like a commercial mail-order operation, with paid employees, office space, and so forth. The costs for all this overhead would have to be added to the price of the items being purchased. A local or regional club might not get quite the quantity discounts a national operation could muster, but a local club uses the donated time of a volunteer doing it at home. That's a hard act to beat, price-wise.

But this still leaves too many important tasks that could be done best by a national organization, which raises the issue: of which ones should be selected for openers. Clearly, it would be much better to select a few of the most significant ones and do them well than to botch a lot of good ideas by taking on too many.

My personal choices for openers would be hardware and software products evaluation and ombudsman services, with a supporting role in a wider effort to achieve computerfest schedule coordination. I prefer these over other possibilities because they are sorely needed, they are particularly appropriate for a single national organization, and they are outward-looking in that they address the amateurs' interaction with the industry. Many of the other aspects desirable in a national organization will evolve naturally and others can be developed as adequate capability is achieved.

There are at least two successful models on which to pattern a product evaluation function: the Consumers Union and the Underwriters' Laboratory. The primary difference lies in the source of their financial support, with Consumers Union financed by its members and publications, while Underwriters' Lab is industry-supported. Both do laboratory testing of products, although Consumers Union must also depend on surveys of users. For both, the independence and integrity of their work is an essential ingredient in their effectiveness.



John and Wayne Loofbourrow talk to the New Jersey Amateur Computer Club

I think that both amateurs and the industry would be willing to provide the necessary support for a comparable service in the small computer field, although either group alone would probably be sufficient. An ombudsman service could fit in neatly, brokering complaints and remedies more efficiently than the separate parties are able to do individually. Should it be decided to pursue these objectives, the next step would seem to be a survey of prospective participants to ascertain interest and support.

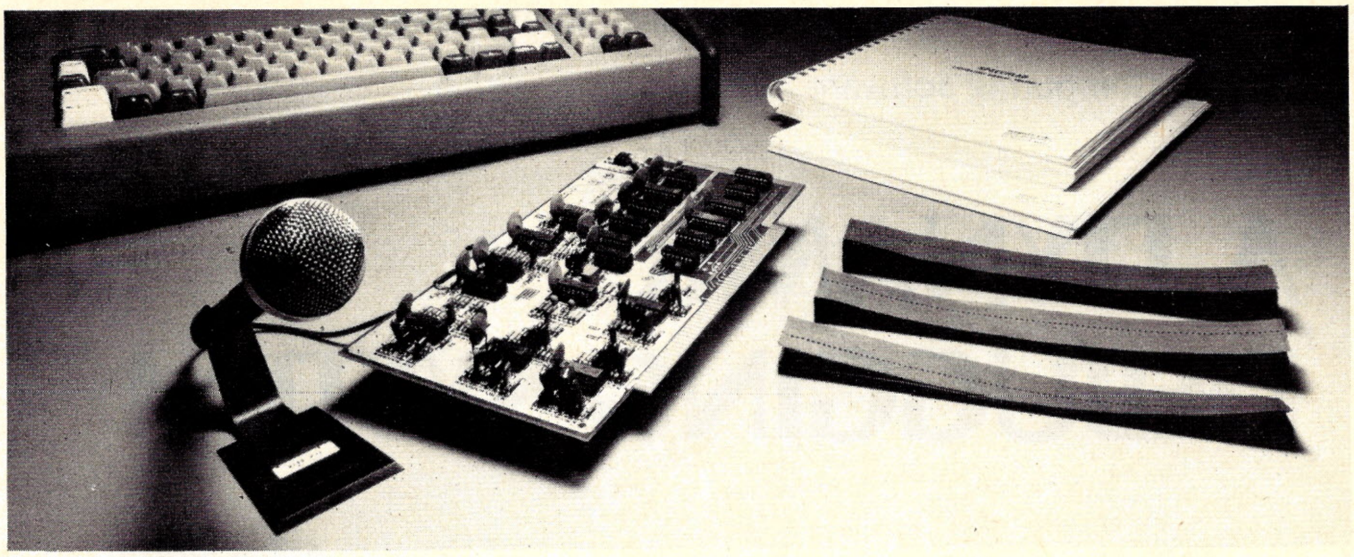
Finally, computerfest scheduling is shamefully uncoordinated despite widespread recognition of the problem. The amateur is distressed by the conflicts that arise, but the exhibitor is really put in a bind to attend the too many shows, some overlapping, with the time, personnel and money they require. While all of us share in the problems, this is one where I believe the companies that are exhibiting will have to take the lead. From the discussions at the Congress I think it's fair to say that the amateurs and their clubs will support any reasonable effort at coordination in this area.

A Matter of Form

Defining a national amateur computer society will naturally have to include some decisions about the form or structure of the new society. There has been a lot of discussion about the basic unit of membership. Should the society be an organization of clubs or an organization of individuals? This is still an open question, but I sense that the weight of opinion is gathering in favor of an organization of individuals. For one thing, a few of the clubs represented at the Congress pointed out that because they were part of a school or company, the club itself could not join although its members could as individuals. There was also concern for the individuals too dispersed to have more than a one-person club. Most influential, perhaps, was the fact that many other national organizations are based on individual memberships with local or regional chapters providing accessible activities.

Unlike other organizations, however, we already have local clubs and regional organizations of clubs, and are only now getting around to forming a national organization. In a sense, the structure of the national society has actually been established and is just waiting to be discovered and incorporated in the definition being developed. An interesting idea proposed at the Congress is that there should be classes of membership which would allow for both individual and club memberships. That would also allow companies to join as sponsoring or institutional members and provide a means for them to support independent product testing.

Three factors suggest that different classes of membership hold the solution to the question of form. First, whatever is decided will have to fit with the organizations that already exist. Second, it will have to be compatible with the functions or purposes to be performed. And third, it must be flexible enough to be able to adjust to new situations and new functions that haven't been thought of yet. ■



Talk to your computer for $299 with SpeechLab.™

Use SpeechLab to directly control any S-100 Bus Computer such as Sol, IMSAI, Altair and so on. SpeechLab can teach you almost as much as the Bell Laboratories know about voice recognition, voice control and computer input.

SpeechLab digitizes and extracts data from speech wave form and applies pattern matching techniques to recognize the vocal input. Response is real time. The system features 64 bytes of storage per spoken word and can handle up to a 64 word vocabulary. And recognition after very little practice is 95 percent or better.

$299* assembled and tested

When we talk price everybody's skeptical. And why not? We give you a complete hardware/software system, a 275 page laboratory manual, 95 page hardware manual and high fidelity microphone.

The lab manual includes 35 graded experiments with over 100 tables and graphs. In fact, it's the only introductory volume on speech recognition currently available.

Software includes SpeechBasic Basic programming language in source and paper tape, assembly language speech recognition program in source and paper tape, hardware self-test program in source and paper tape. SpeechBasic plot, correlation, recognition and advanced recognition programs are offered in source.

Hard to believe, you bet. True? A Los Angeles customer says, "I love your kit!!! I have 40 boards and 2 IMSAI's and your kit was the best documented of them all. I love the way you integrated the software and hardware together. I love your lab manual."

We loved those comments. They tell the story better than we ever could. The LA customer did ask who the founders of the firm were so he could relate better. They're a couple of gifted young engineers who got tired of the big firm, big technology trip and decided to take a chance with a better idea.

You can't get better quality
You can't get more performance

Sure, more complex, higher price equipment is available for about 50 times more money. It won't do much more than you can do with SpeechLab. And the quality and state-of-the-art engineering can't be any better. We use CMOS design for low power and ultimate reliability.

See SpeechLab at your nearby computer store

Selected computer stores have SpeechLab on display. Visit your nearest. If he doesn't have it, ask him to contact us or simply write us directly.

Heuristics, Inc., Box B, 900 N. San Antonio Road
Los Altos, CA 94022, Phone (415) 948-2542

☐ Send me SpeechLab. I enclose $299.
California residents add sales tax.

☐ Send me more information.

Name ____________________

Street ____________________

City ____________ State ________ Zip ________

In Canada:
Trintronics Ltd.
186 Queen St. W.
Toronto, Canada M5V 1Z1

Heuristics INC.

*Available in kit form for slightly less.

A Creative Computing Software Report......

A DYNAMIC DEBUGGING SYSTEM

Steve North

Are your eyes weary from staring at blinking lights as you attempt to debug a program the hard way? Or are you fed up with the tedium of using a simple debugging monitor with only breakpoints to help you? If you own an 8080-based microcomputer, then relief is in sight. Relief, in this case, is the Dynamic Debugger, written by David Benevy and marketed by the Computer Mart of New Jersey.

The Dynamic Debugging System (DDS) is a software debugging facility for 8080 assembly-language programs. DDS is actually a program which helps you to debug your own program. A simple debugging monitor lets you enter breakpoints in your program, runs your program for awhile, and then lets you see the condition of the registers and program when the breakpoint was hit. DDS, on the other hand, monitors the execution of your program as it is going on, displaying pertinent information and identifying possible problem areas. It uses a softcopy device for the display (versions are available for the Processor Technology VDM-1 and popular CRTs, but the VDM version is better, since information may be updated almost instantaneously. DDS can only be used on CRTs with cursor addressing.) A keyboard is used to communicate with DDS.

The Dynamic Debugger has three types of functions. The first type controls the display of the program being debugged. Actually, most of the display is handled automatically. At the middle of the top of the screen, DDS displays the current program counter and the next five instructions to be executed, in easy-to-read mnemonic format. To the right of this is a display of all the CPU registers, the words they point to, both as hex and ASCII characters. The individual bits in the flag word are also broken up into mnemonics. In the 64-column version they are displayed as single characters (C for carry, for instance). In the 80-column version, the entire name of the flag (such as CARRY) is displayed if the flag is on. To the left side of the top of the screen, DDS displays the current stack pointer, the top five words on the stack, and the words they point to, as hex and ASCII characters. DSS also maintains its own stack of valid return addresses, which is displayed to the far left of the screen. So at a glance you can see exactly what's going on in all the registers and the stack, and have a pretty good idea what your program is doing too.

Now for the part of the display which the user controls. There are two types of displays you may ask for. One of these is a hex/ASCII display of selected portions of memory. There are six lines of display available in the 16-line version, and 11 lines in the 24-line version. These lines may be split up as you wish. For instance, lines 1 and 2 could display memory from 3000 to 301F, while lines 2 through 6 display memory from 2000 to 203F. You may also display memory as instructions instead of raw data. Two such groups of instructions may be displayed simultaneously. The instructions are displayed in mnemonic format similar to the display of the code currently being executed at the top of the screen. When selected, these displays are located in the lower half of the screen. These displays are requested by DM (display memory) and DI (display instruction) commands. The two bottom lines on the screen are used for displaying error messages and echoing keyboard input, respectively.

The second group of commands is used for manipulating the user's program or other data. Using the keyboard, you may enter bytes, words (groups of two bytes), or characters into memory; fill portions of memory with a constant; move blocks of memory, or find a combination of up to three bytes. You may also modify any of the registers or register pairs including the program counter and stack pointer, the stack itself, or the Dynamic Debugger's separate stack of return addresses. So you also have complete control over everything going on in your computer.

The third and most powerful set of commands is used to control and execute the program to be debugged. When debugging a program with DDS, your program never actually takes control of the system. Instead, DDS executes your program step by step, updating the display as often as you wish. (This is why it's nice to use a VDM-1. The VDM can be updated very quickly but CRTs may take a while longer.) As your program runs, you can see the program counter advance, registers change, data get pushed on and popped off the stack, or portions of memory change.

Stepping

One character on the keyboard (ESCAPE) is used as the stepping character. Every time this key is entered, part of the user's program is executed by the DDS. The exact number of instructions executed depends on the value you enter with the STEP command. By entering a step of one, the program may be single-stepped by hitting the stepping character. Or you may enter a larger step and run longer sections of code.

You may also set the time delay between instructions, which permits you to execute the program slowly while referring to a source listing. Execution of your program stops when DDS executes the specified number of instructions or when you enter a key (it doesn't matter which one).

Detection

The Dynamic Debugger gets even better: it not only tells you what's going on, it stops when it detects something wrong. For instance, if you PUSHed more data on the stack than you POPped in a subroutine, there would normally be an invalid return address on the stack and your program could run off into the woods. By maintaining its own list of valid return addresses, DDS will stop when it thinks you're trying to use a bad return address. Since some programmers do tricky things with the stack such as intentionally pushing extra words on the stack to set up return addresses, this feature may be disabled. DDS also halts if you attempt to execute an undefined opcode, a HLT, or a RST.

Error Condition

DDS also permits you to enter your own error conditions. For instance, you can tell DDS to stop executing your program if the program references data outside of a certain range, if the program reaches a specified address, if certain opcodes are executed, if the stack goes outside of a given range, if the program attempts to store in a certain location, etc. Note that for most of these, several conditions may be active at once. For instance, up to five opcodes may be used in the opcode stop option. When DDS finds an error, it displays an error message on the error line and stops executing your program.

```
CALL  STK  SP=3FF9     ADDR  ---INST----- OP    REGS
----  ------------     -------------------- --  ---------------
2AB5  2AB5=E604 ..     3000  CALL    CAB4 CD    AF=00FF MZAPC
2388  2388=0418 ..     3003  JMP     0040 C3    BC=03DB=CD18 ..
1234  1234=02B2 ..     3006  SBB   E        9B  DE=2310=8303 ..
      FF00=FFFF ..     3007  INR   C        0C  HL=2320=C00F ..
      FFFF=C3FF ..     3008  POP   H        E1

2000  MVI   A,02   3E              200D  SHLD    2330 22
2002  STA     2300 32              2010  SHLD    232F 22
2005  LXI   D,2310 11              2013  STA     2400 32
2008  STAX  D       12             2016  DCR   A      3D
2009  LXI   H,2320 21              2017  JNZ     2002 C2
200C  MOV   M,A    77              201A  MVI   B,02   06
                                                 COPYRIGHT 1977
=>GO,D000.                                     COMPUTER MART NJ
```

DDS display with DI (Display Instructions) command.

Since DDS executes about 300 instructions per second maximum, a standard breakpoint facility is available for running sections of code that are lengthy or critical about timing.

Evaluation

This is an extremely powerful software debugging tool. What more do you need for debugging programs? At all times you know the status of almost everything inside your computer and if your program has done something foolish. DDS might even be a handy tool for teaching the inner workings of the 8080 (provided it wasn't overwhelmed by DDS).

Anyway, I couldn't think of anything critical to say about DDS, so Larry Stein (owner of the Computer Mart of NJ) mentioned a few questions people have asked about the Dynamic Debugger.

```
CALL  STK  SP=3FF9     ADDR  ---INST----- OP    REGS
----  ------------     -------------------- --  ---------------
2AB5  2AB5=E604 ..     3000  CALL    CAB4 CD    AF=00FF MZAPC
2388  2388=0418 ..     3003  JMP     0040 C3    BC=03DB=CD18 ..
1234  1234=02B2 ..     3006  SBB   E        9B  DE=2310=8303 ..
      FF00=FFFF ..     3007  INR   C        0C  HL=2320=C00F ..
      FFFF=C3FF ..     3008  POP   H        E1

0100 0000 00C3 F000 C900 00C3 CF01 CD1F C0CA ................
1000 CA2B 10C6 4032 DA0F 4E79 E61F C640 32DB .+..@2..Ny...@2.
3000 CDB4 CAC3 4000 9B0C E122 297D 31A1 2939 ....@....")}1.)9
3010 1A40 CA9E 5888 D85A 233B A97B 3BEB FFFF .@..X..Z#;.{;...
3020 081A 9D9E 9C1E 5B05 3F39 35AB F1A1 A93A ......[.?95....:
2455 E981 02A7 85B5 AFA5 ED18 A383 5469 80C5 ............Ti..
                                                COPYRIGHT 1977
=>GO,D000.                                    COMPUTER MART NJ
```

Dynamic Debugging System display with DM (Display Memory) option in use.

(1) Can it debug Z-80 programs? Well, sort of. When DDS reaches Z-80 code, it will display an II (Illegal Instruction) message and wait. You can then continue execution if you want. DDS doesn't display all the Z-80 registers, either. So, this version of the Dynamic Debugger really isn't designed for Z-80s. It's not for IBM-370's either, says Larry.

(2) What happens if you attempt to do an input from the keyboard which DDS is also using? Obviously that won't work out too well, because DDS is constantly checking the keyboard, since it stops whenever a key is pressed. The solution is to move into the accumulator what you were going to input. This is only a problem if you have only one keyboard in your system.

(3) What happens when your program executes code in ROM (for instance, an I/O routine)? While code in ROM is being executed, the DDS is inactive. It takes over as soon as control returns to code in RAM. If the code in ROM doesn't return to the instruction following the call, a breakpoint will have to be inserted. This shouldn't be much of a problem since programs in ROM are usually already debugged.

(4) How do you debug programs that use the VDM-1 for output? That might be a little tricky. The DDS does permit you not to refresh the screen after every instruction, and has a clear command which completely rewrites the screen. Useful if your program puts some junk on the screen. That's about the best you could ask for, since you're trying to debug and output with the same device. If you won't like it, you're welcome to your blinking lights!

Anyway, I consider most of these to be rather trivial questions.

Dynamic Debugging System Commands

AR	Address range stop request
AS	Address stop request
BK	Run at full speed until breakpoint is reached
CALL	Pushes a word on DDS's CALL stack
CLR	Completely rewrites the screen
DI	Display memory as instructions in mnemonic format
DM	Display memory as hex and ASCII characters
DY	Set time delay between instructions
EB	Enter bytes into memory
EC	Enter characters into memory
EP	Enter program counter
ER	Enter register
ERP	Enter register pair
ES	Enter stack pointer
EW	Enter word into memory
FILL	Fill block of memory with a constant
FIND	Find a series of bytes in memory
GO	Same as BK except that the breakpoint is not retained
MOVE	Move blocks of memory
OS	Opcode stop request
POP	Pop a word from the stack
PUSH	Push a word on the stack
REF	Controls whether the DI or DM display is automatically updated
RET	Removes a word from the CALL stack
RS	Reference stop request
SR	Stack range stop request
SS	Store stop request
ST	Sets number of steps executed by stepping character
TC	Controls use of automatic return address validation
VSR	Value stop on register request
VSB	Value stop on memory byte request
VSP	Value stop of register pair request
VSW	Value stop of memory word request

Note: Most of these commands require arguments which are entered with the command, separated by commas. A period indicates the end of a command.

The DDS is available for $30 on CUTS or Tarbell cassette, and $35 on iCOM floppy disk, from the Computer Mart of New Jersey, 501 Route 27, Iselin, NJ 08830. It will be provided with a relocating loader (another nifty idea) which will let you stuff DDS anywhere in your system's memory. DDS is 4K in size. Complete details on interfacing to your system are provided. Despite my proclivity for finding problems in almost any product, this is one I can recommend without hesitation. Get one if you do any 8080 assembly-language programming, or just like to hack around with fancy software! ■

AM-100

You have to SEE it to BELIEVE it!

The Alpha Microsystems AM-100 is LIGHT YEARS ahead of everything else you've seen so far in the low cost computing field.

For a FRACTION of what you'd normally pay for the SOFTWARE ALONE, you get a 16-bit processor with ALL of these BIG-SYSTEM capabilities:

MULTI-TASKING, MULTI-USER TIMESHARING

- ☆ DEVICE INDEPENDENT I/O
- ☆ ADVANCED FILE STRUCTURE
- ☆ POWERFUL SYSTEM COMMANDS
- ☆ SOPHISTICATED TEXT EDITOR
- ☆ FULL MACRO ASSEMBLER
- ☆ LINE PRINTER SPOOLER
- ☆ RE-ENTRANT, MULTI-USER BASIC *COMPILER*
- ☆ LARGE UTILITIES LIBRARY

Yet, with all this it's still compatible with the S-100 BUS!

If you like the Decsystem-10 operating system, if you like TECO . . . if you like the PDP-11 instruction set . . . you'll LOVE the AM-100!

ONLY
$1495
IN STOCK *NOW!*

NOW AT

BYTE SHOP

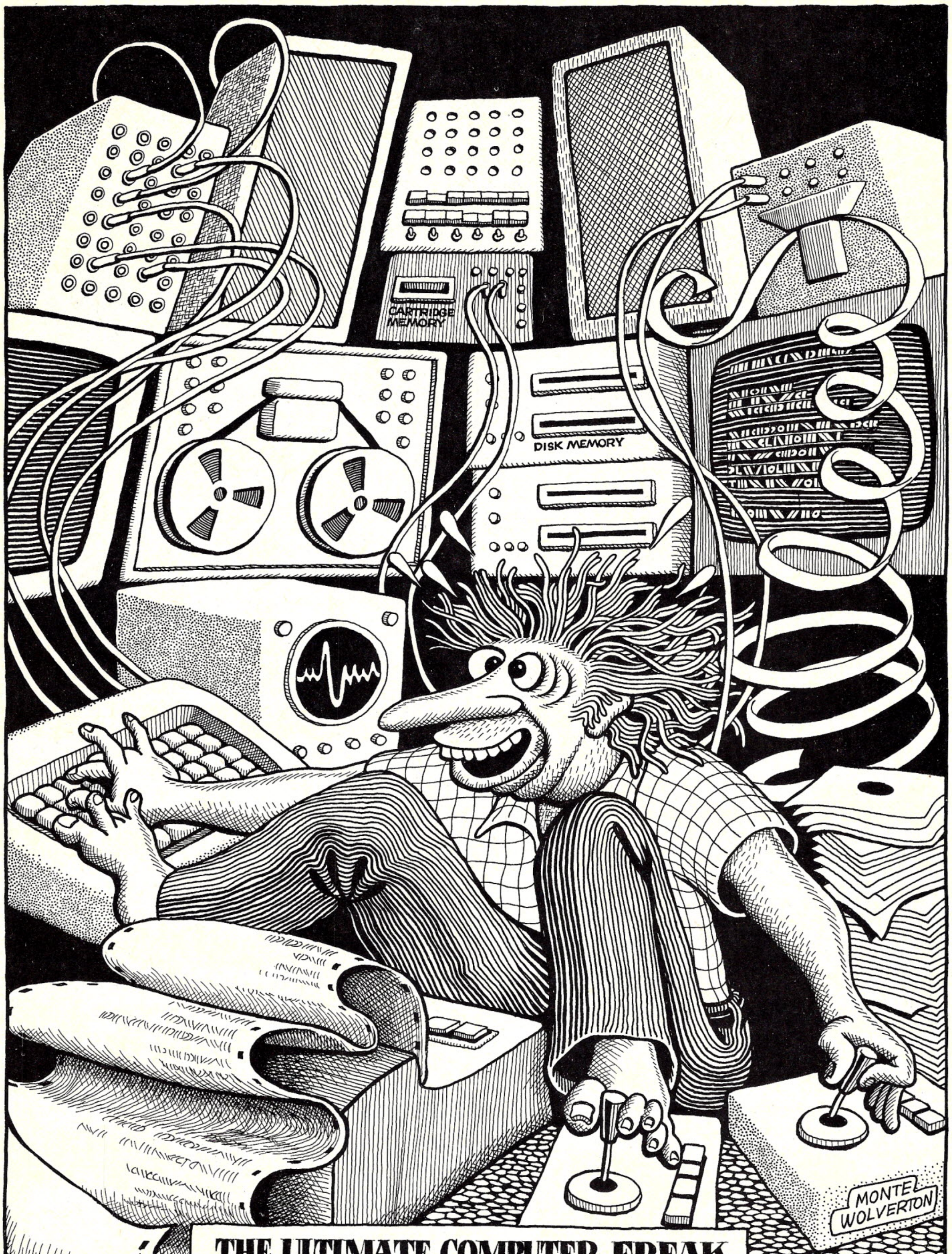
of Pasadena

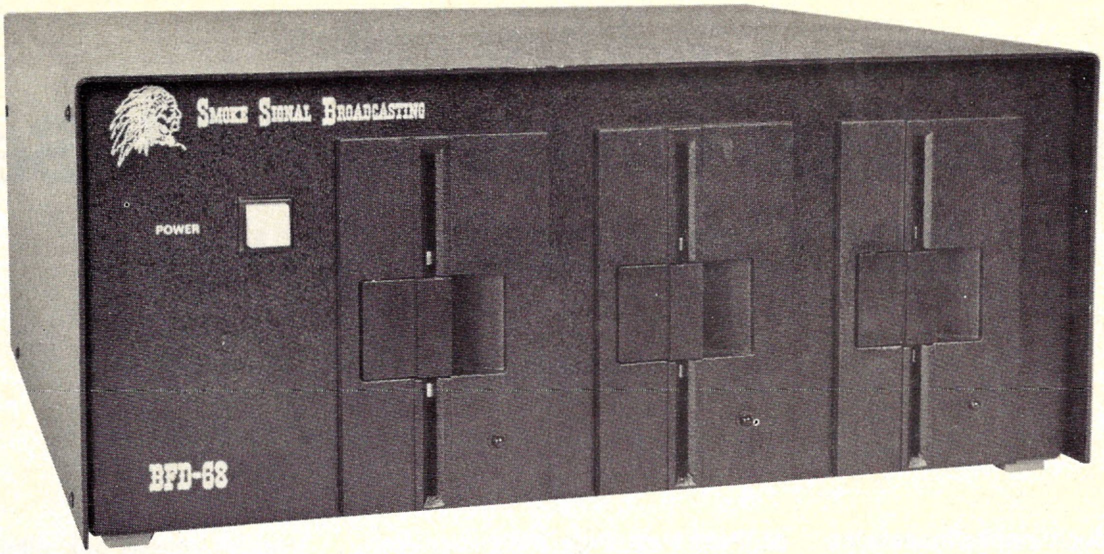
496 S. LAKE AVE.
PASADENA, CA. 91101
PHONE: (213) 684-3311

HOURS: Tuesday — Friday, 12:00 — 9:00;
Saturday & Sunday, 12:00 — 5:00;
Closed Mondays

master charge THE INTERBANK CARD

BANKAMERICARD


CARTRIDGE MEMORY
DISK MEMORY
MONTE WOLVERTON
THE ULTIMATE COMPUTER FREAK



- ANY NUMBER OF FILES MAY BE OPEN (IN USE) AT ONE TIME
- THE NUMBER OF FILES AND SIZE OF FILES IS LIMITED ONLY BY THE SIZE OF THE DISC
- MERGING FILES REQUIRES NO EXTRA DISC SPACE
- NO WAITING FOR THE DISC TO RE-PACK
- LONGER DISC LIFE — MORE EVEN DISC WEAR

Our Basic Floppy Disc System (BFD-68) must, in all modesty be called superb. It comes completely assembled with a disc controller that is plug compatible with the SWTPC 6800. In fact all our products use the 6800 standard SS-50 (Smoke Signal 50) bus used by SWTPC. The cabinet and power supply are capable of handling up to 3 Shugart Mini-Floppy Drives. One drive is included in the price of the BFD-68 and others may be added easily at any time. Or you may save money by ordering the dual-drive BFD-68-2 or triple drive BFD-68-3 (pictured). Price: BFD-68 $795, BFD-68-2 $1139, BFD-68-3 $1479, SA-400 Drive $360.

The BFD-68 includes our Disc Operating System Software. Our software provides for a soft-sectored disc format consisting of 128 bytes per sector, 18 sectors per track and 35 tracks per disc. The software provides direct commands to name and rename files, transfer memory to disc and disc to memory and to automatically jump to the starting location of any program loaded from disc to memory. The direct command names are: RUN, GET, GETHEX, CLOSE, SAVE, DELETE, APPEND, RENAME, COPY, LIST, FIND, LINK and PRINT. In addition, the Disc File Management subroutines are available to create files under your program control.

A bootstrap PROM is included on the controller board to initiate the Disc Operating System which loads into a 4K memory board located at 7000 or optionally at D000. Thus, you can be up and running from a cold start in just a few seconds.

SUPER SOFTWARE

Free patches are provided for SWTPC BASIC version 2.0 and Co-Resident Editor/Assembler. These patches allow the SAVE and LOAD commands to work with the disc or the cassette at your option.

SUPER EDITOR: Smoke Signal Broadcasting now has its own editor. It is a content oriented editor with string search and block move capability. Changes may be made by referring either to line number or string content or a combination of references. Naturally, it is designed for file transfers to and from the BFD-68. Price: SE-1 $29 on diskette or cassette.

ALL OUR PRODUCTS EXCEPT THE PS-1 ARE COMPLETELY ASSEMBLED.

SUPER ASSEMBLER: Inputs source code from file on the BFD-68 disc system and outputs object code to disc file. Assembly listings include alphabetized and tabulated symbol table. Price: SA-1 $29 on diskette or cassette.

Complete source listing included for both editor and assembler. Order both for $53 and save $5.

SMARTBUG — A CURE FOR MIKBUGITIS: A super smart Motorola-Mikbug replacement that preserves almost all Mikbug entry locations so your present programs will run without modification. Uses ACIA instead of PIA and includes many additional features including a software single-step trace command. A SMARTBUG listing is included and object code is provided on a 2708 free with each P-38 series board purchased. Source listing available separately for $19.50.

NEED A FULL SIZE FLOPPY? Our P-38-FF is a plug-in interface card to the ICOM Frugal Floppy™. It includes all the features of the P-38-I plus one 2708 EPROM containing the ICOM bootstrap software. Just plug the P-38-FF into your SWTPC 6800 and your ICOM into the P-38-FF and you're ready to use the Frugal Floppy and ICOM's 6800 software package. Price $299.

Our P-38 is an 8K EPROM board containing room for 8 2708's. Or, you may use it to hold up to 7 2708's plus your Motorola Mikbug or Minibug II ROM. The P-38 addressing is switch selectable to any 8K location. Price $179.

The P-38-I contains all the features of the P-38 plus an interface to the Oliver Paper Tape Reader and our EPROM Programmer. Price $229.

The PS-1 Power Supply Kit provides plus and minus 16 volts required for the P-38 series boards. Also, it allows a wiring modification to be made to the 8 volt supply that will increase its output by one volt. Price $24.95.

Our M-16-A is a 16K single power supply STATIC RAM memory system. The M-16-A is fully buffered and requires only half the power of a similar size system using low-power 2102's. With the M-16-A, you can expand your system to 48K and still have room left for one of our EPROM boards. The M-16-A is switch selectable to any 4K starting address and hardware write protect is included. Quick delivery. Price $529.

BANKAMERICARD, VISA AND MASTER CHARGE WELCOME.

SMOKE SIGNAL BROADCASTING

P.O. Box 2017, Hollywood, CA 90028 • (213) 462-5652

Some Tips On Using A TV Set For Computer Output

David H. Ahl

So you've bought yourself a spanking new SOL-20 that expects to provide output to a video monitor. Or you have a VDM or TV Dazzler of Apple-II or Poly 88, etc., etc., all of which put out a beautiful video signal for a video monitor. But unfortunately you don't have a video monitor.

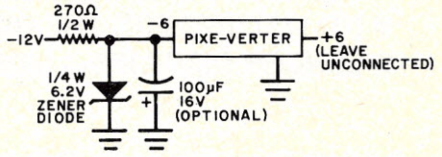
In this situation you have three choices:

1. *Modify a standard B&W TV* to act as a video monitor. Generally this can be done fairly easily by breaking the existing signal between the video detector and video amplifier stages and then inserting the new signal into the input of the video-amplifier stage. You may have to add one or more capacitors, diodes, and/or transistors to preserve the white and sync levels. If you want to still be able to use the set as a tv, you'll want a SPDT switch at the break in the circuit. It is *highly* advisable to use a transformer-type tv set with the chassis isolated from the power line. If you follow this route, you'll find the appropriate circuits in DonLancaster's book, *TV Typewriter Cookbook* published by Howard W. Sams, or in Don's article "Television Interface" in the October 1975 issue of *Byte.*

If you don't want to brew up the circuit yourself, Pickles & Trout makes a kit (TVM-04) for modifying a Hitachi TV set with the SX chassis (Models P-03, P-04, P-05, P-08, P-53, P-63, etc.) into a quality video monitor. The TVM-04 can accept a 0.6V P-P video signal as from a Polymorphics VTI or 1.4 V P-P as from a Processor Technology VDM. The kit contains a switch permitting use as either a monitor or TV set. The TVM-04 is available for $20 postpaid from Pickles & Trout, P.O. Box 2276, Goleta, CA 93018. In the New York area, a 12-inch Hitachi P-05 TV set sells for about $80, thus the total cost is $100 or so.

2. *Purchase a TV Monitor.* A good-quality B&W 9-inch Hitachi monitor (VM-909) sells for about $185. Larger-screen Comrac, RCA, or Sylvania industrial-quality monitors are in the $500 price range. Super quality if you can justify the expense.

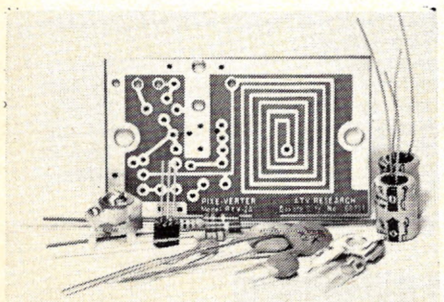
3. *Build an RF Modulator.* This device accepts a video signal and adds an RF carrier so that the signal can simply be fed into the antenna terminals of any standard TV set. For color output (from the TV Dazzler) this seems to be the best approach, since "monitoring" a color set is risky and color monitors are expensive. One RF modulator, the PXV-2A Pixe-Verter requires -6V (actually anything between -5 and -6.5 V) which can be furnished by batteries or from the computer. The device is so small (1.25" x 2.1" x 0.8") that it will fit inside the computer practically anywhere. Be sure it is inside the computer or in a shielded chassis box, otherwise your computer will be broadcasting to your whole neighborhood. Most computers have available either -12 or -16 volts of filtered DC. The following circuit will allow you to power a Pixe-Verter from this.



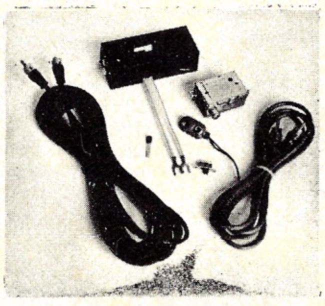
Change the dropping resistor to 470 ohms for -16 volts. In this set-up, the Pixe-Verter draws about 2 ma and the Zener diode, 20 ma. The Pixe-Verter is available for $8.50 postpaid from ATV Research, 13th and Broadway, Dakota City, NB 68731.

A similar device with some added goodies is available from M&R Enterprises. This one has an antenna/modulator selector switch, long cables, and operates from a wide voltage range (6-12 VDC). This RF Modulator is available for $25.95 postpaid from M&R Enterprises, P.O. Box 61011, Sunnyvale, CA 94088.

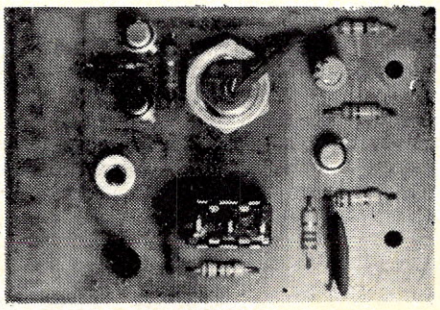
Another source of interest for those hard-to-find video cables, connectors, adaptors, etc. (as well as audio plugs, cables, etc.) is WIDL Video, 5325 North Lincoln Ave., Chicago, IL 60625. phone (312) 271-4629. ■



Pixe-Verter before assembly



M&R RF modulator kit



Pickles & Trout monitor kit

IF YOU'RE NOT SUBSCRIBING TO CREATIVE COMPUTING, YOU'RE NOT GETTING THE MOST OUT OF YOUR COMPUTER.

No computer magazine gives you more applications than we do! Games. Puzzles. Sports simulations. CAI. Computer art. Artificial intelligence. Needlepoint. Music and speech synthesis. Investment analysis. You name it. We've got it. And that's just the beginning!

Whatever your access to computer power—home computer kit, mini, time-sharing terminal—*Creative Computing* is on your wavelength. Whatever your computer application—recreation, education, business, household management, even building control—*Creative Computing* speaks your language.

Read through pages of thoroughly documented programs with complete listings and sample runs. All made easy for you to use. Learn about everything from new software to microprocessors to new uses for home computers. And all in simple, understandable terms. And there's still more. *Creative Computing* discusses creative programming techniques like sort algorithms, shuffling and string manipulation to make your own programming easier and more efficient.

We can even save you time and money. Our extensive resource section is filled with all kinds of facts plus evaluations of hundreds of items. Including microcomputers, terminals, peripherals, software packages, periodicals, booklets and dealers. We also give you no-nonsense equipment profiles to help you decide which computer is best for you—before you spend money on one that isn't.

We've got fiction too. From the best authors in the field, like Asimov, Pohl and Clarke. Plus timely reviews of computer books, vendor manuals and government pamphlets. And so much more!

Isn't it about time you subscribed to *Creative Computing*? It's the smart way to get the most out of your computer.

Complete this coupon and mail it today. Or for fast response, call our toll-free hot line. (800) 631-8112. (In New Jersey call (201) 540-0445).

I want to get the most out of my computer.
Please enter my subscription to:

creative computing

Term	USA	Foreign Surface	Foreign Air
□ 1-year	□ $ 8	□ $ 12	□ $ 20
□ 2-year	□ $ 15	□ $ 23	□ $ 39
□ 3-year	□ $ 21	□ $ 33	□ $ 57
□ Lifetime	□ $300	□ $400	□ $600
□ Vol. 1 Bound	□ $ 10	□ $ 12	□ $ 15
□ Vol. 2 Bound	□ $ 10	□ $ 12	□ $ 15

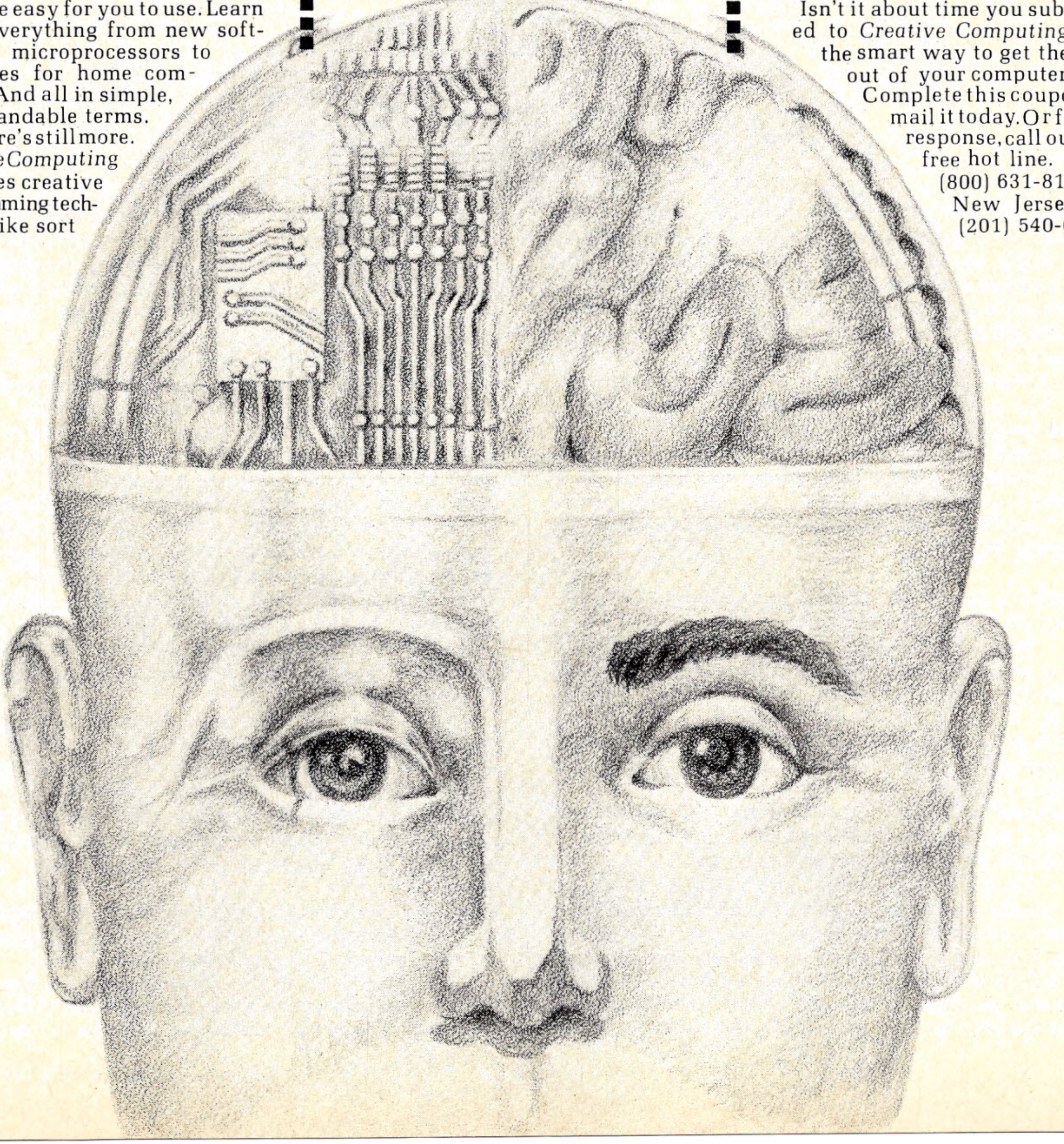
□ New □ Renewal □ Gift
□ Visa/Bank Americard □ Master Charge
Card No. ______________
□ Please bill me ($1.00 billing fee will be added; foreign orders must be prepaid)

Name ______________

Address ______________

City __________ State ______ Zip ______

Send to: Creative Computing, Attn: Alyce
P.O. Box 789-M, Morristown, N.J. 07960



World Model Bibliography

Large-scale dynamic global computer models have predicted everything from doomsday by 2050 to complete equilibrium. On a smaller scale, models tend to be more consistent and hence more useful. Nevertheless, global models are getting better and there's an increasing exchange of information between rival groups. Dennis Meadows of Dartmouth, project leader of the original MIT study ("The Limits to Growth" was the outcome), has prepared a bibliography of sources of interest to people involved with global modeling.

The layman will also be interested in two other articles, "Doomsday, Says MIT Computer, May Be Just 100 Years Away," and "Dynamic Modeling Using Fortran IV," both of which appeared in *The Best of Creative Computing, Vol. 1.*

1. Forrester, Jay W. *World Dynamics.* Cambridge, MA: Wright-Allen Press, Inc., 1971.

   World2, the model on which *World Dynamics* is based, was the first significant global model. It was conceived and constructed by Professor Jay Forrester at the M.I.T. Sloan School of Management to serve as the basis for a two-week seminar on global problems. Model equations are presented in the book in the DYNAMO language.

2. Meadows, Donella H.; Meadows, Dennis L.; Randers, Jorgen; and Behrens, William W., III. *The Limits to Growth.* New York: Universe Books, 1972.

   Limits was the result of a two-year effort by ten scientists and students at M.I.T. under the direction of Dennis and Donella Meadows to extend and disaggregate the model originally presented by Jay Forrester. While the resulting publication provides much more empirical data in support of the general thesis, the conclusions remain essentially the same as those first sketched out in *World Dynamics.* This text presents a comprehensive verbal description of the important causal mechanisms governing growth and collapse, but it does not present the complete equations required to reproduce the detailed results. Those are included in reference 4 below.

3. Meadows, Dennis L., and Meadows, Donella H., eds. *Toward Global Equilibrium: Collected Papers.* Cambridge, MA: Wright-Allen Press, Inc., 1973.

   This book is an unedited collection of thirteen papers, each commissioned to explore a specific aspect of the Limits to Growth project at M.I.T. There are two introductory papers by Forrester and Meadows, seven papers describing specific models of population, agricultural, or resource issues, and four reports that explore the implications of growth limits for different institutions. The seven modeling papers each contain detailed computer listings in the DYNAMO language.

4. Meadows, Dennis L. et al. *Dynamics of Growth in a Finite World.* Cambridge, MA: Wright-Allen Press, Inc., 1974.

   This is the technical report resulting from the Limits to Growth effort at M.I.T. It contains detailed technical expositions on each of the five world model sectors: population, resources, agriculture, pollution, and capital investment. Extensive references to the sources of theories incorporated in the model and to data used in estimating its coefficients are also provided.

5. Mesarovic, Mihajlo, and Pestel, Eduard. *Mankind at the Turning Point.* New York: E.P. Dutton and Co., Inc., Reader's Digest Press, 1974.

   Mankind at the Turning Point is the general report prepared from results obtained by two teams working jointly at Hannover, Germany, and Case

Western Reserve University in Cleveland, Ohio. The basic conclusions of *Limits* are reaffirmed in this text, but the regional disaggregation employed by Mesarovic and Pestel permits more detailed exploration of regional interactions than could be done with the M.I.T. models. This report does not include the equations necessary to reproduce its runs nor are they available in any other form.

5a. Mesarovic, Mihajlo, and Pestel, Eduard. *Multilevel Computer Model of World Development System.* 6 vols. Luxembourg: International Institute for Applied Systems Analysis, 1974.

This six-volume report set was released in conjunction with a presentation of the Mesarovic-Pestel model before the staff of the International Institute for Applied Systems Analysis, but the complicated input/output software of the model prevents even the information in the detailed exposition from being implemented.

6. Linneman, Hans. *Population Doubling and Food Supply.* Amsterdam: Economic and Social Institute of the Free University, 1974.

The effort of this Dutch team was undertaken in order to determine the investment and social implications of a doubling world population. To narrow the focus of the study, exclusive concern was given to the question of nutrition. The analysis was based in part on simulation models, the equations for which are not given in this document.

7. Herrera, Amilcar O. et al. *Catastrophe or New Society? A Latin American World Model.* Ottawa: International Development Research Centre, 1976.

The Bariloche model was undertaken by a team of social scientists at the Bariloche Foundation in South America with support from the International Development Research Council in Canada. Initially conceied as a criticism and rebuttal of the Limits to Growth thesis, this model is essentially normative in concept and provides some description of those policies which would have to be employed to meet the basic needs of the world's population by the year 2000.

8. Leontieff, Wassily; Carter, Anne; and Petri, Peter. *The Future of the World Economy: A Study on the Impact of Prospective Economic Issues and Policies on the International Development Strategy.* New York: United Nations, 1976.

The Leontieff model was commissioned by the United Nations. It employs an input/output matrix with exogenous population variables and a rather scanty resource and pollution sector to examine interregional trade flows over the next several decades.

9. Kahn, Herman; Brown, William; and Martel, Leon. *The Next 200 Years.* New York: William Morrow and Company, Inc., 1976.

This volume is part of an ongoing study by Herman Kahn and his associates within the Hudson Institute to chart out some areas of the future. The work is not based on any formal computer model, nor are the detailed underlying assumptions leading to the authors' conclusions actually specified or incorporated into the text.

10. Tinbergen, Jan. *Reshaping the International Order.* Edited by Antony J. Dolman. New York: E.P. Dutton and Co., Inc., 1976.

RIO is the description of a study undertaken to investigate narrowing the gap between the rich and the poor countries of the world. Although the organizers did not use a formal computer model, they did develop, through meetings of international committees, several schemes for economic equity. The thrust of the study is distribution and self-sufficiency. ■

Part II in this series looks at FOR...NEXT, TAB, and hotdogs

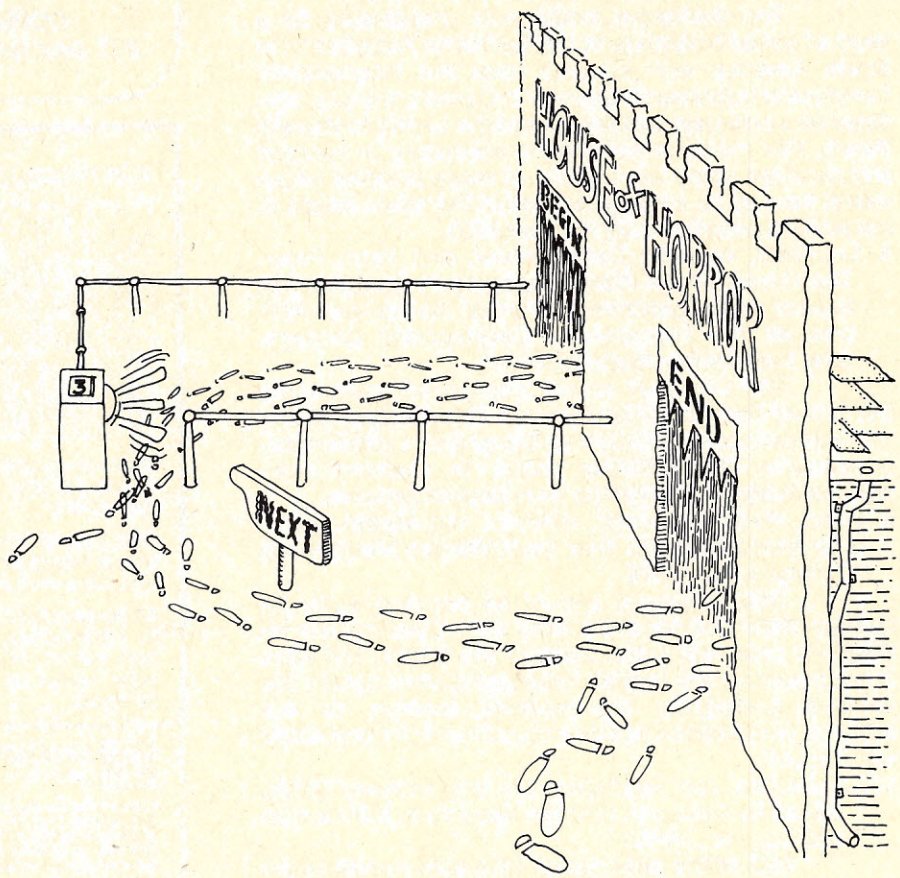
THE 8-HOUR WONDER

All About BASIC Programming in One Long Day (or Eight Short Nights)

Thomas A. Dwyer

2.3 HOUR 3: GETTING THE COMPUTER TO DO ITS OWN COUNTING

In the last section we showed the technique of using a counter together with an IF...THEN statement to control how many times a program executes a group of statements. This is called "looping" or "iteration," and it's an important type of control in programs. There is another way to control looping that is even simpler. It uses a pair of statements: a FOR statement together with a matching NEXT statement.



FOR. . .NEXT

Here are two programs that compare the two techniques for controlling loops:

Using a Counter

```
10 LET K = 1
20 IF K    5 THEN 60
30 PRINT K; K*K; K*K*K
40 LET K = K + 1
50 GO TO 20
60 END
```

Using FOR and NEXT Statements

```
10 FOR K = 1 TO 5
20 PRINT K; K*K; K*K*K
30 NEXT K
40 END
```

Both programs produce the same output:

```
RUN

1   1    1
2   4    8
3   9    27
4   16   64
5   25   125
```

As you can see, the second program is simpler. Here's another example showing how several statements (called the *body* of the loop) can be controlled by FOR...NEXT statements:

```
LIST

10 FOR K=1 TO 3
15 PRINT "LOOP # ";K
20 PRINT "TYPE A NUMBER";
30 INPUT N
40 PRINT "THE CUBE OF YOUR NUMBER IS";N*N*N
50 PRINT
60 NEXT K
70 END


RUN

LOOP #  1
TYPE A NUMBER? 5
THE CUBE OF YOUR NUMBER IS 125

LOOP #  2
TYPE A NUMBER? 15
THE CUBE OF YOUR NUMBER IS 3375

LOOP #  3
TYPE A NUMBER? -3
THE CUBE OF YOUR NUMBER IS-27
```

THIS IS THE BODY OF THE LOOP.

THE "PRINT" IN LINE 50 CREATES A BLANK LINE.

The full form of the FOR statement is

```
100 FOR K = 1 TO 25 STEP 5
  -----(body of the loop)-----
200 NEXT K
```

The FOR statement really has three key words: FOR, TO, and STEP. The word STEP is used to say how much K should be incremented each time around the loop. If STEP is omitted, the STEP size (or increment) is taken to be 1.

Here's an example to show a negative STEP:

```
LIST

10 PRINT "STAND BY FOR AIR TIME"
20 FOR K=5 TO 1 STEP -1
30 PRINT K;"SECONDS"
40 NEXT K
50 PRINT "YOU'RE ON!!"
60 END


RUN

STAND BY FOR AIR TIME
 5 SECONDS
 4 SECONDS
 3 SECONDS
 2 SECONDS
 1 SECONDS
YOU'RE ON!!
```

An important feature of the FOR statement is that variables or arithmetic expressions can be used after the = sign, and also after TO and STEP.

Here's a simple example showing this feature:

```
LIST

10 PRINT "HOW MANY STARS DO YOU WANT TO BE PRINTED";
20 INPUT N
30 FOR K=1 TO 2*N
40 PRINT "*";
50 NEXT K
60 PRINT
70 PRINT "HA HA--THAT'S TWICE AS MANY AS YOU WANTED."
80 END


RUN

HOW MANY STARS DO YOU WANT TO BE PRINTED? 5
**********
HA HA--THAT'S TWICE AS MANY AS YOU WANTED.
```

SELF-TEST

1. Simulate running this program and write down the output you get.

```
10 PRINT "IF JAN 1 IS A MONDAY THEN"
20 FOR K = 1 TO 31 STEP 7
30 PRINT "JAN"; K; "IS A MONDAY"
40 NEXT K
50 END
```

2. Simulate running this program and write down the output.

```
10 LET N = 10
20 FOR K = 1 TO N STEP N/5
30 PRINT K
40 NEXT K
50 END
```

3. Modify the MULTIPLICATION PROGRAM of Section 2.2 so that the number of problems done is controlled by a FOR...NEXT loop instead of the K counter.

2.4 HOUR 4: PRINTING PATTERNS; THE HOTDOG PROBLEM

Let's start by reminding ourselves of how to use a semicolon to keep printing on the same line, and how to use a PRINT to "undo" the effect of this semicolon. Look at the difference betweenthese two programs:

```
10 FOR K = 1 TO 5            10 FOR K = 1 TO 5
20 PRINT "*";                20 PRINT "*";
30 NEXT K                    30 NEXT K
40 PRINT "FINISHED"          40 PRINT
50 END                       50 PRINT "FINISHED"
                             60 END
RUN
                             RUN
*****FINISHED
                             *****

                             FINISHED
```

In the second program, the PRINT in line 40 was needed to get a line feed and carriage return so that FINISHED appeared on a new line.

Now let's get fancy, and use two FOR loops, one inside the other. The second loop acts like the *body* of the first, and we say we have *nested* FOR loops.

```
LIST

10 FOR L=1 TO 3
20 PRINT "OUTER LOOP HAS L =";L
30 FOR N=1 TO 5
40 PRINT "*";
50 NEXT N
60 PRINT
70 NEXT L
80 END

RUN

OUTER LOOP HAS L = 1
*****
OUTER LOOP HAS L = 2
*****
OUTER LOOP HAS L = 3
*****
```

OUTER LOOP

INNER LOOP

INNER LOOP PRODUCED 5 ASTERISKS.

If you think through this program, you'll see that the body of the inner loop (which is simply line 40) gets executed 15 times. Looking at the asterisks printed should make this clear. The variable L controls how many lines get printed (3), while N controls how many asterisks per line (5), so 15 are printed altogether.

Could we have nested, nested FOR loops? You bet. Here's an example where N controls the number of asterisks per line, L controls how many lines, and B controls how many blocks of lines.

```
LIST

10 FOR B=1 TO 3
20 PRINT "B =";B
30 FOR L=1 TO 4
40 FOR N=1 TO 15
50 PRINT "*";
60 NEXT N
65 PRINT
70 NEXT L
80 NEXT B
90 END
```

```
RUN

B = 1
***************
***************
***************
***************
B = 2
***************
***************
***************
***************
B = 3
***************
***************
***************
***************
```

Here's a trickier version of the above which you should study carefully to make sure you understand what's going on.

```
LIST

10 FOR B=1 TO 3
20 FOR L=1 TO 2*B
30 FOR N=1 TO 2*(L+B)
40 PRINT "*";
50 NEXT N
60 PRINT
70 NEXT L
75 PRINT
80 NEXT B
90 END

RUN

****
******

******
********
**********
************

********
**********
************
**************
****************
******************
```

THIS TABLE SHOWS WHAT'S HAPPENING:

B GIVES THE BLOCK NUMBER.

L GIVES THE LINE NUMBER.

N GIVES THE NUMBER OF ASTERISKS.

B	L	N
1	1	4
1	2	6
2	1	6
2	2	8
2	3	10
2	4	12
3	1	8
3	2	10
3	3	12
3	4	14
3	5	16
3	6	18

NOTE: THESE ARE THE CONTENTS OF B, L, AND N JUST AFTER PRINTING A LINE OF ASTERISKS.

We'll return to the subject of printing patterns later, and show how to make them more interesting by using random numbers and other tricks.

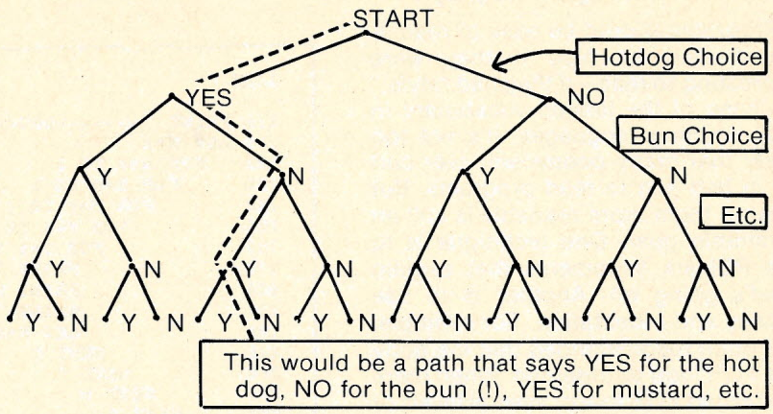
Let's switch to another use of nested loops by showing an application to a fun problem which is also related to the important idea of *tree structures*.

THE HOTDOG PROBLEM

Suppose you're running the hotdog stand at your next club picnic, and you decide to post a computer printout showing how to order all the possible combinations by number. Let's assume that there are only YES/NO decisions allowed for hotdog, bun, mustard, mayonnaise, and catsup. To discourage overindulgence, we'll also

print a calorie count for each combination.

The way to think about this problem is to picture what's called a *decision tree*.



One way to generate a tree structure in BASIC is to use nested FOR loops, one for each level. Our tree will have five levels (one for each ingredient) so there will be five FOR loops. Here's how all the paths through our five-level tree can be tabulated with a BASIC program.

```
LIST

10 PRINT "          DOG     BUN    MUST.    MAYO.   CATSUP"
15 LET K=1
20 FOR H = 0 TO 1
30 FOR B = 0 TO 1
40 FOR M = 0 TO 1
50 FOR Y = 0 TO 1
60 FOR C = 0 TO 1
70 PRINT "#";K;":  ";
80 PRINT H;"    ";B;"    ";M;"    ";Y;"    ";C;
90 PRINT "     CALORIES=";H*140+B*120+M*20+Y*100+C*30
95 LET K=K+1
100 NEXT C
110 NEXT Y
120 NEXT M
130 NEXT B
140 NEXT H
150 END

RUN
            DOG     BUN    MUST.    MAYO.   CATSUP
# 1 :        0       0       0       0       0     CALORIES= 0
# 2 :        0       0       0       0       1     CALORIES= 30
# 3 :        0       0       0       1       0     CALORIES= 100
# 4 :        0       0       0       1       1     CALORIES= 130
# 5 :        0       0       1       0       0     CALORIES= 20
# 6 :        0       0       1       0       1     CALORIES= 50
# 7 :        0       0       1       1       0     CALORIES= 120
# 8 :        0       0       1       1       1     CALORIES= 150
# 9 :        0       1       0       0       0     CALORIES= 120
# 10 :        0       1       0       0       1     CALORIES= 150
# 11 :        0       1       0       1       0     CALORIES= 220
# 12 :        0       1       0       1       1     CALORIES= 250
# 13 :        0       1       1       0       0     CALORIES= 140
```

WEIGHT WATCHER'S SPECIAL

The output from this program would be a lot nicer if all the 0's and 1's (NO and YES decisions) lined up. We can make this happen by using the new key word TAB.

TAB

The statement 10 PRINT TAB (12); "*" will cause the "*" to print in column 12 (don't forget that columns are numbered from left to right starting with *zero*). We'll say more about TAB in chapter 3, and show how using the form TAB (X) (where X is a variable in your program) can be used to produce graphical output.

To fix up our hotdog problem, all we have to do is change one line.

```
70 PRINT "#";K;TAB(5)":   ";

RUN
            DOG     BUN    MUST.    MAYO.   CATSUP
# 1  :       0       0       0       0       0      CALORIES= 0
# 2  :       0       0       0       0       1      CALORIES= 30
# 3  :       0       0       0       1       0      CALORIES= 100
# 4  :       0       0       0       1       1      CALORIES= 130
# 5  :       0       0       1       0       0      CALORIES= 20
# 6  :       0       0       1       0       1      CALORIES= 50
# 7  :       0       0       1       1       0      CALORIES= 120
# 8  :       0       0       1       1       1      CALORIES= 150
# 9  :       0       1       0       0       0      CALORIES= 120
# 10 :       0       1       0       0       1      CALORIES= 150
# 11 :       0       1       0       1       0      CALORIES= 220
# 12 :       0       1       0       1       1      CALORIES= 250
# 13 :       0       1       1       0       0      CALORIES= 140
# 14 :       0       1       1       0       1      CALORIES= 170
```

A Word About Programming Style

Programs should be easy to read. It would also be nice if they were interesting to read—if they had "style." Because of the limited vocabulary in programming languages, it's not too likely that many people will ever curl up in bed just to read programs. But making them more readable is still an admirable goal. One technique is to use REMark statements that explain what's going on. Another is to use spaces and indentation. For example, it is often recommended that the body of a FOR loop be indented. When there are nested FOR loops, several levels of indentation are needed. This can sometimes get to be more confusing than helpful, as seen in the following "stylistic" version of the Hotdog program. Also note that it's *very* hard to type.

```
LIST

110 PRINT "-------DOG----BUN---MUST.---MAYO.--CATSUP"
115 LET K=1
120   FOR H=0 TO 1
130     FOR B=0 TO 1
140       FOR M=0 TO 1
150         FOR Y=0 TO 1
160           FOR C=0 TO 1
170             PRINT "#";K;TAB(5);":    ";
180             PRINT H;"     "B;"      ";M;"      ";Y;"      ";C;
190             PRINT "    CALORIES=";H*140+B*120+M*20+Y*100+C*30
195             LET K=K+1
200           NEXT C
210         NEXT Y
220       NEXT M
230     NEXT B
240   NEXT H
250 END
```

Another "catch" to using lots of spaces and REMark statements is that they increase the size of programs, and also slow down their execution. So you'll find many microcomputer programmers going in the opposite direction, and writing things like this:

10FORX=1TON:PRINTX:NEXT

In this book we'll try to avoid both extremes. We'll also work at improving readability by using "balloons," "brackets," and other extra notations that don't hurt execution since they're not part of the program. More complicated programs will be broken into segments, and these will be distinguished by REMark statements with easily-spotted dashed lines. For an example of the "balloon" technique, look at the plot program in section 7.2. For an example of the dashed-line REMark technique, see the horse-race program in section 6.5.

An example of using external brackets to distinguish the nested FOR loops in the HOTDOG problem is as follows. This approach avoids the memory/speed/typing problems of extra spaces within the program.

THESE FIVE LOOPS RUN THROUGH ALL POSSIBLE CHOICES OF INGREDIENTS.

DOG LOOP
BUN LOOP
MUSTARD LOOP
MAYO. LOOP
CATSUP LOOP

```
LIST

10 PRINT "          DOG     BUN    MUST.    MAYO.  CATSUP"
15 LET K=1
20 FOR H = 0 TO 1
30 FOR B = 0 TO 1
40 FOR M = 0 TO 1
50 FOR Y = 0 TO 1
60 FOR C = 0 TO 1
70 PRINT "#";K;":   ";
80 PRINT H;"     ";B;"      ";M;"      ";Y;"      ";C;
90 PRINT "   CALORIES=";H*140+B*120+M*20+Y*100+C*30
95 LET K=K+1
100 NEXT C
110 NEXT Y
120 NEXT M
130 NEXT B
140 NEXT H
150 END
```

THIS IS THE BODY OF THE LOOPS. IT PRINTS ONE LINE EACH TIME IT'S EXECUTED.

SELF-TEST

1. Simulate a run of this program:

```
10 FOR B = 1 TO 2
20 FOR L = 3 TO 1 STEP -1
30 FOR N = 1 TO B*L
40 PRINT "*";
50 NEXT N
60 PRINT
70 NEXT L
80 PRINT
90 NEXT B
100 END
```

2. Write and run a "hotdog" program that allows a triple meat choice of no-dog, beef frank, or kielbasa.

3. Write a program that uses nested FOR loops to print the multiplication tables for 7, 8, and 9.

   Here's a start:

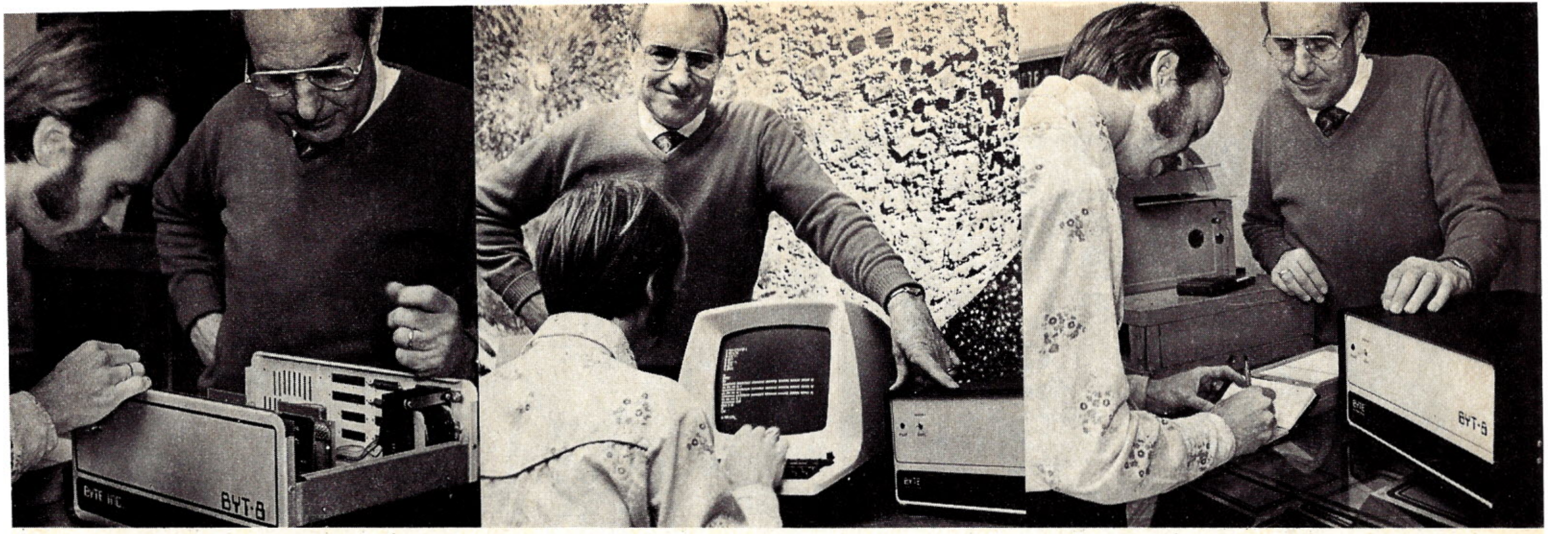
```
10 FOR T = 7 TO 9
20 FOR K = 0 TO 12
30 PRINT K; "TIMES"; T; "="; K*T
40 ... etc....
```

4. (Optional) Read ahead to the chapter on strings, and see if you can make the hotdog program print words instead of numbers so the lines of output look like this:

   #28 : DOG BUN MAYO. CATSUP CALORIES = 390

NEXT ISSUE

Part III of this exciting series looks at READ . . . DATA, RESTORE, INT, ON . . . GOTO . . . and the fantastic RND. Don't miss it! ■



The Byte Shop is *Ready, Willing and Able* to satisfy your computing needs.

Ready

Your Byte Shop is ready to supply computer products off the shelf for the computer hobbiest, experimenter, home entertainer, or business entrepreneur. Computers in kit form or assembled and tested.

Willing

Your Byte Shop is willing to help in the advice, education, or handholding necessary for your selection of a computer tailored for your application.

Able

Your Byte Shop is able to provide the full service and support required for a product of sophistication.

Why

Because Byte Shop wants to be your Computer Store.

What are you waiting for? Come on in.

Arizona
Phoenix-East
813 N. Scottsdale Rd.
Phoenix-West
12654 N. 28th Drive
Tucson
2612 E. Broadway

California
Berkeley
1514 University Ave.
Burbank
1812 W. Burbank Blvd.
Campbell
2626 Union Ave.
Diablo Valley
2989 N. Main St.
Fresno
3139 E. McKinley Ave.
Hayward
1122 "B" Street
Lawndale
16508 Hawthorne Blvd.
Long Beach
5433 E. Stearns St.
Mountain View
1063 W. El Camino Real
Palo Alto
2233 El Camino Real
Pasadena
496 W. Lake Ave.
Placentia
123 E. Yorba Linda
Sacramento
6041 Greenback Lane
San Diego
8250 Vickers-II
San Fernando Valley
18424 Ventura Blvd.
San Francisco
321 Pacific Ave.
Santa Barbara
4 West Mission
Stockton
7910 N. Eldorado St.
Thousand Oaks
2707 Thousand Oaks Blvd.
Ventura
2409 Main St.
Westminster
14300 Beach Blvd.

Colorado
Arapahoe County
3464 S. Acoma St.
Boulder
2040 30th St.

Florida
Cocoa Beach
1325 N. Atlantic Ave., Suite 4
Ft. Lauderdale
1044 E. Oakland Park Blvd.
Miami
7825 Bird Road

Minnesota
Eagan
1434 Yankee Doodle Rd.

New York
Levittown
2721 Hempstead Turnpike

Ohio
Rocky River
19524 Center Ridge Rd.

Oregon
Beaverton
3482 SW Cedar Hills Blvd.
Portland
2033 SW 4th

Pennsylvania
Bryn Mawr
1045 W. Lancaster Ave.

South Carolina
Columbia
2018 Green St.

Utah
Salt Lake City
261 S. State St.

Washington
Bellevue
14701 NE 20th Ave.

Canada
Winnipeg
665 Century St.

Japan
Tokyo
Towa Bldg., 1-5-9
Sotokanda

BYTE SHOP
the affordable computer store™

TO BE CONTINUED...

We're the memory house

Associated Electronics Company
1885 W. Commonwealth, Unit G
Fullerton, CA 92633
(714) 879 - 7541 879 - 7707

ORDER FORM

	QTY. KIT	ASSEMBLED
AE 8KS	☐	☐
AE 16 KPS	☐	☐
AE 1702A PRGM	☐	☐
BROCHURE	☐	

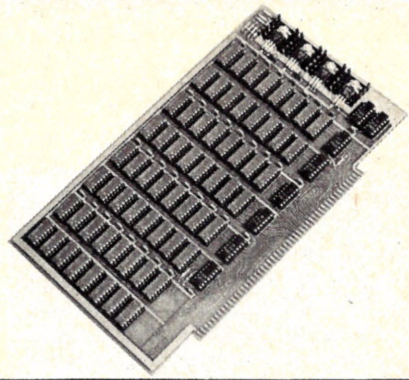
Name: ______________________

Address: ______________________

____

We accept: Checks • Money Orders • C.O.D. • Credit Cards
Master Charge #______________
BankAmericard #______________

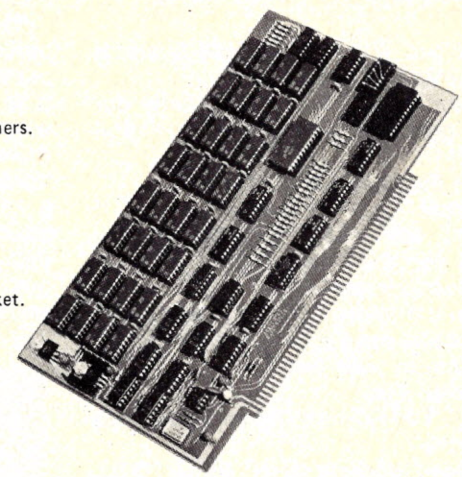
AE 8 KS (8K STATIC RAM)



- 8192 bytes of static memory.
- Access time: 500 nanoseconds.
- Memory chips are Signetics 21L02's.
- Memory allocation is in 8K increments to any location (switch selectable).
- Memory protection (full 8K) selectable with a switch.
- Fully socketed and buffered.
- Solder mask and character screen for ease of assembly.

Kit $189.00 Assembled $289.00

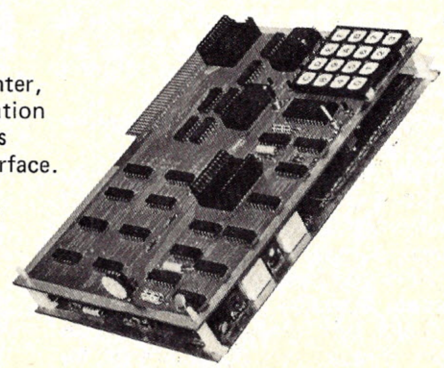
AE 16 KPS (16K PSEUDO-STATIC RAM)



- Runs with all popular processors including Polymorphics, Sol, Chromemco, ZPU, and others.
- Runs with Micropolis, North Star, Tarbell and other mass storage systems.
- Full speed memory chips (500 nanoseconds) with no wait states.
- All memory management functions on board. No software management required.
- Low power consumption.... 2.5 to 5 watts maximum!!!
- Headers allow allocating memory in 4K blocks to ANY LOCATION within the 65K 8080 or Z80 memory space. Allocation must be on 4K boundaries.
- Phantom, memory protect-unprotect, and other functions jumper selectable.
- This memory card is as DMA compatible as any other pseudo-static memory on the market. The DMA controller MUST HONOR ALL PROCESSOR SIGNALS!!! ASK DYNABYTE which DMA controllers they run with... then ask us!!!!
- Fully socketed and buffered.

Kit $349.00 Assembled $449.00

AE 1702A PRGM (MANUAL 1702 PROGRAMMER)



Our new EPROM programmer has a vast array of improvements over the original design... with NO INCREASE IN PRICE!!!
THE ORIGINAL DESIGN has sold to John F. Kennedy Space Flight Center, Jet Propulsion Labs, Los Alamos Scientific Lab, Stanford Linear Acceleration Center, IBM, XEROX, 3M, and many others. The new design incorporates on-board 3 state buffers and two 16 pin headers for direct processor interface.

The new design also incorporates a new keyboard for increased reliability and improved function selection.

Write for additional details.

Kit $189.00 Assembled $299.00

An interview with Richard Peterson,

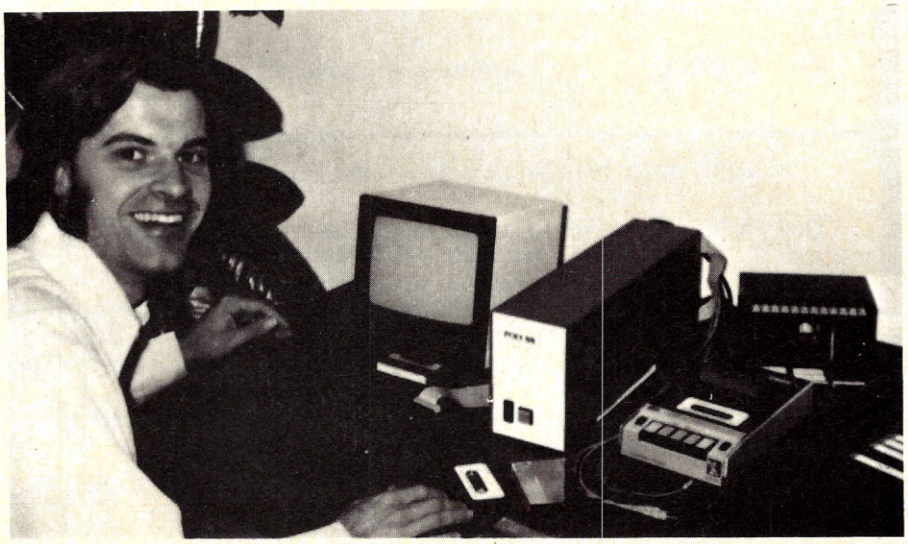
president of PolyMorphic Systems.

"I don't like noisy machines"

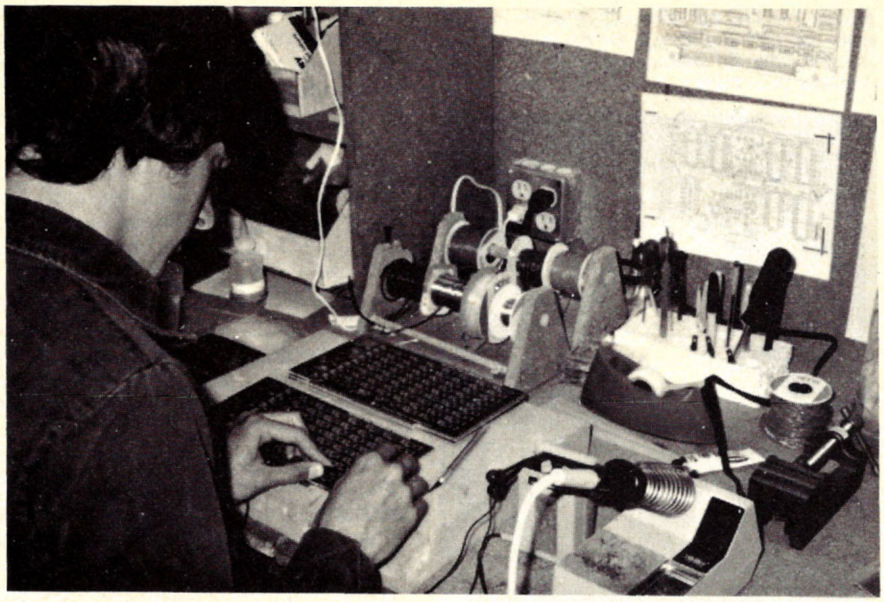
Nursing the after-effects of two carafes (yes carafes) of margaritas the night before, Sandy and I spent a delightful morning with Richard Peterson, the genial but astute president and founder of PolyMorphic Systems. What follows is a loosely edited transcript of portions of our conversation.-DHA

Ahl: When did Polymorphic Systems come into existence?
Peterson: November 1975, with our first product, an analog-to-digital board.
Ahl: Is that still being offered?
Peterson: Yes. We sell a few but it doesn't sell like our video-display board for the Altair bus, which we introduced in mid-1976.
Ahl: And that provided the real growth and capital base so you could produce a system of your own?
Peterson: Yes.
Sandy: Today, how many people do you have?
Peterson: Working here? About 20-25.
Sandy: Do you do everything here?
Peterson: Pretty much. We just moved here, actually. We've only been here about two months. We got so crowded where we were that everything was on top of everything else and we couldn't get anything done.
Ahl: What was your background before you got into this?
Peterson: I did research with a small consulting company. Mostly contract work for the Dept. of Transportation. We used computers, of course.
Ahl: Were you in computer science?
Peterson: Actually, I was in physics.
Ahl: Let's talk about your Poly 88 system. It looks very different from an Altair or IMSAI which has lights and switches all over the front panel. There's just one little "on" button and a reset light. How do you toggle in a loader?
Peterson: Our people here were mostly interested in using computers and the lights didn't buy us anything. In fact, they just confuse people and also they add cost to the machine so we just cut them out.

Because we know where all the peripherals are (display, keyboard, cassette, etc.), we've got a thousand-byte monitor ROM which comes up when you turn the power on. It's always there, right on the CPU board. It has a loader for two formats of tape including a 2400-baud Poly tape and the 300-baud Byte format. We load files by name into the machine. You turn the machine on and type "LOAD BASIC" and it comes back and says "BASIC IS READY" and load a program in Basic. You then refer to file names; you can store all your files contiguously on a tape and not have to search and try to find out where you are in each one.
Ahl: I think it's very clever, as you're loading a program on the Poly 88, to have the display tell you every 256 words that you're still loading properly.
Peterson: And that the check sum is right and that everything is going O.K. You don't have to wait until a long file is all loaded to know if you have a problem.
Ahl: Would your version of Basic run on another 8080 computer or is it specifically designed for this one?
Peterson: It's not really designed to work on other systems; I think it wouldn't run very well. It requires a 60-cycle real-time clock on the CPU, which is in the Poly 88. One of the commands we have in Basic is time.



Richard Peterson, president of PolyMorphic Systems, with a complete Poly 88 system (computer, keyboard, TV monitor, cassette recorder).



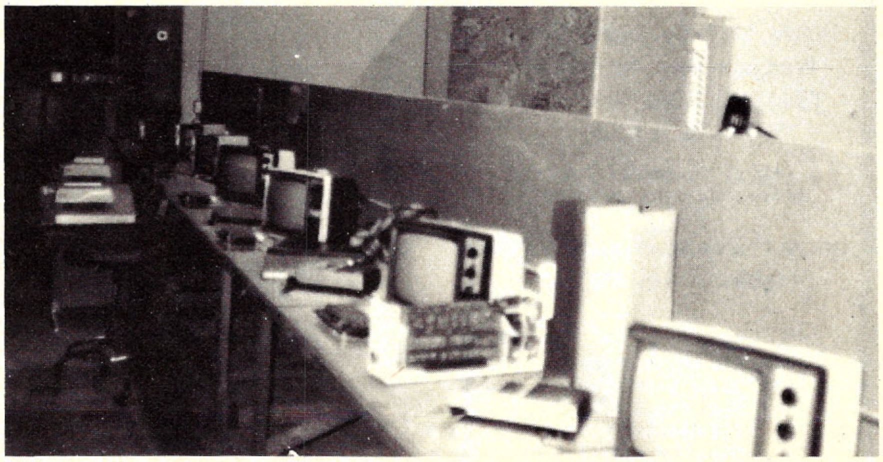
A technician at the beginning of assembling a board.

You can just use a function called TIME; you find out how many seconds it was since the last time you called TIME and so you have a time base.
Ahl: That's very handy for writing games and for a lot of things; CAI, for example, where you can find out how long it took before somebody typed in their response.
Peterson: Sure.

Our CPU board has a lot more on it than most CPU boards; that's how we get it in such a small box. We have a 1K ROM; a small USART board to which you can connect both a cassette and a serial port, a similar small board for a printer, 512 words of RAM, vectored interrupt, and a real-time clock.
Ahl: What about memory? How much do you have in this system?
Peterson: We have two 8K static RAM boards; 16K total.
Ahl: Now the display picture is in memory, right?
Peterson: The display is contained in 1K on the video board which is just like any other memory to the processor. It's just that the memory happens to be displayed.
Ahl: The Poly 88 has five slots; how do you see them being used?
Peterson: I see this as the popular configuration for someone who wants to play with software: 1 slot, CPU; 2 slots, 8K memory each; 2 slots left over for whatever you want to do. That's really what our system is designed for. It's designed to get up and running very easily, to write program tapes and be able to distribute them to other people and that kind of thing.
Ahl: What kind of people do you find getting them mostly?
Peterson: Well, all kinds of different groups. The largest group is people who want a system in their home just to play with, and learn about.
Sandy: Is this system purchased already assembled, or is it sold in kit form?
Peterson: Yes, both kit and assembled. Customer's choice. You can start out without any memory, you can start out with just the CPU board and video board and buy your own memory and peripherals if you want.
Ahl: For a starter system what would the CPU, housing and power supply cost?
Peterson: About $700. That includes the memory that's on the CPU board and on the video board. It doesn't include any mass memory. Everything together with 16K of memory, monitor, keyboard and cassette recorder is about $2000. Assembled.
Sandy: How did you make the system so quiet? Is it because the fan isn't on?
Peterson: I don't like noisy machines. I don't think machines should have to be noisy. We don't have huge power supplies and huge overkill which you need with other computers. Our CPU board has everything on it that takes four full boards in an Altair or IMSAI. So that's one of the reasons why we get away with it; it's smaller.

As a matter of fact we're packaging it all in a big box; the computer, a keyboard, cassette recorder, TV set, manuals — the whole kit. On the other hand, I don't feel that we're going to be able to compete terribly well with something like the Fairchild game. Those games sell for under $200; they offer incredible things that I'm not sure even the Dazzler can compete with because of the lower resolution — at least currently. However, the Fairchild game has it, but more than that — it's cheap. We can't compete with that at all.

Our selling point is the fact that you can see the game, you can see how the game was constructed in Basic, and you know what makes this thing operate. You can understand the machine, whereas the Fairchild game and the others are just black boxes and you jam a tape (or ROM) into a slot and it takes off. You don't have any understanding, whereas this is a learning tool.
Sandy: You said there are lots of other applications besides games — what are some of the other things?
Peterson: Well, for these computers to sell they have to do things useful for people. It's like cars; you know, sixty years ago, people bought cars and showed them off but until cars were useful and they had roads to use them on they didn't sell very well. But there are a lot of useful things these machines can do today. I'm not sure using it for your home budget is useful; however, analysis of stocks and bonds — investigating whether you should do this or that requires an incredible amount of work by hand — you can say if this happens then that will happen and this is what my financial status will look like. I think too, they're certainly valuable in businesses.
Ahl: What about things like text editing?



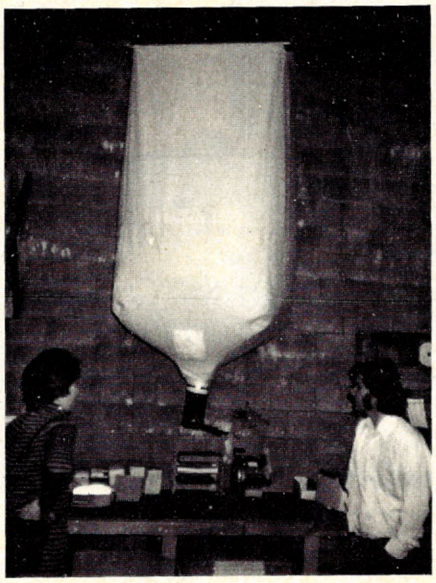
Poly 88 systems are "burned in" and tested before shipping.

Peterson: That's a fantastic use of this machine — text editing.
Ahl: Do you have the software?
Peterson: We're developing it. I think that the weak spot right today is in the cassette. However, our floppy disc system will be out shortly. [Pub. note: it's out today.] A small floppy can fit 30 pages on it. And it's not too hard to pull one out and put another in. Also, they're easy to use, especially for people who have never used computers before.

We look also to the educational market. Our Basic has a very good error diagnostic routine that actually points to the error, not "ERROR CODE 3 IN LINE 10" or that kind of stuff. I think this is valuable in learning to use Basic. Also, most schools are already oriented to using Basic. They're not looking for a lot of assembly-language programming or loading in through front-panel switches or that kind of stuff. I would think schools would hate that kind of thing; it can cause a lot of frustration. That's what's happening with schools right now with punch cards. The kids get so frustrated just getting their cards and a printout half an hour later saying it doesn't work and then trying to find someone to explain to them what their printout means.

Also, compared to timesharing, it's great because if one kid bogs the system and does something to cause a crash it doesn't affect all the users, just the one who caused it. This is a tremendous advantage.

Sandy and Richard admire PolyMorphic's popcorn machine (the bag holds the puffed plastic packing pieces affectionately known in the industry as popcorn).



Sandy: One thing I've been trying to figure out; if you're not a computer nut and you really don't want to spend a lot of time fooling around with things and trying to make a system run if something goes wrong, how do you get it serviced?
Peterson: You take it back to the computer store you bought it from. I recommend you buy it from a store that's going to guarantee you that kind of support. A lot of these stores are providing good service, some are not, so — you have to be careful.
Ahl: You also have to be careful that the store is reasonably well financed. If you keep track of how many are opening and also how many are closing, you find they're opening at the rate of 3 a week and closing at the rate of one a week.

Where do you see the market going from here: hobbyists, education, business — what do you think it'll be in five years?
Peterson: There are going to be a lot of people selling business accounting systems and stuff like that. There's going to be a lot of people disappointed in that but I see these things as tools, small tools. IBM is building steamshovels; we're building camels. You know we're not making the same thing. But there are a lot of small problems that need solving. CitiBank for example, is decentralizing a lot of their gear. And they're utilizing a lot of these small computers for small problems.

But also the home user will be buying them for things we haven't even thought of today. Also hobbyists. And schools. The future is going to be fantastic in ways we don't even imagine today! ■

ok® wire wrapping center ok®

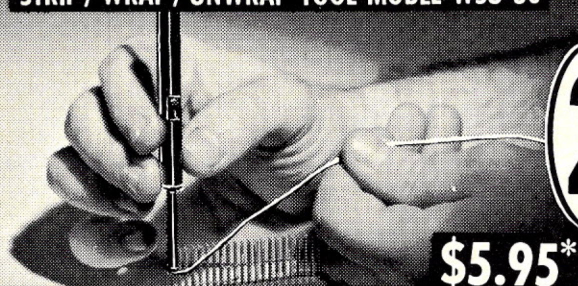
ANOTHER UNIQUE PRODUCT
DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE
BY
OK MACHINE & TOOL CORPORATION

1

BATTERY WIRE-WRAPPING TOOL MODEL BW-630

$34.95*

STRIP / WRAP / UNWRAP TOOL MODEL WSU-30



$5.95*

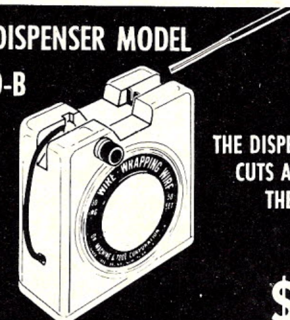
2

ANOTHER UNIQUE PRODUCT
DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE
BY
OK MACHINE & TOOL CORPORATION

ANOTHER UNIQUE PRODUCT
DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE
BY
OK MACHINE & TOOL CORPORATION

3

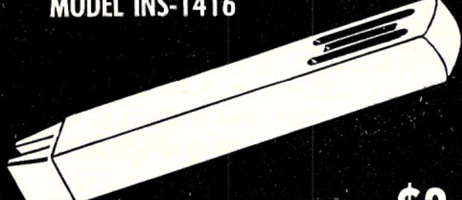
WIRE DISPENSER MODEL
WD-30-B



THE DISPENSER WHICH CUTS AND STRIPS THE WIRE

$3.45*

DIP IC INSERTION TOOL WITH PIN STRAIGHTENER
MODEL INS-1416



$3.49*

4

ANOTHER UNIQUE PRODUCT
DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE
BY
OK MACHINE & TOOL CORPORATION

ANOTHER UNIQUE PRODUCT
DESIGNED, MANUFACTURED AND MARKETED WORLDWIDE
BY
OK MACHINE & TOOL CORPORATION

5 ***WHAT'S ? NEXT***

* MINIMUM ORDER $25.00, SHIPPING CHARGE $1.00, N.Y. CITY AND STATE RESIDENTS ADD TAX

OK MACHINE & TOOL CORPORATION
3455 Conner St., Bronx, N.Y. 10475 ■ (212) 994-6600 ■ Telex 125091

Random, systematic and guided trial-and-error strategies are described in this fourth article in the series

Thinking Strategies with the Computer: Trial-and-Error

Donald T. Piele and Larry E. Wood*

What is the difference between a method and a device? A method is a device which you use twice."
G. Polya

In the course of our formal education, we are taught a great variety of *devices* for solving problems that have already been neatly grouped together at the end of each chapter of a textbook. Typical examples of these devices are formulas, equations, rules, and theorems which are studied for the purpose of attacking certain types of problems. After a careful study of these specific techniques, exams are given to test our ability to recall them. But what do you do when you are faced with the more realistic situation of not being told what device is likely to solve a problem or, worse yet, of having forgotten how to use a technique altogether? Is all hope lost? Of course not, although many students, by the time they reach college, believe that it is.

At the beginning of each semester, we like to ask the students in our freshman and sophomore classes to try to solve a favorite problem of ours by any method they can. It is the Pigs and Chickens problem which appeared in our first article on Thinking Strategies (Piele & Wood, 1977).

PIGS AND CHICKENS

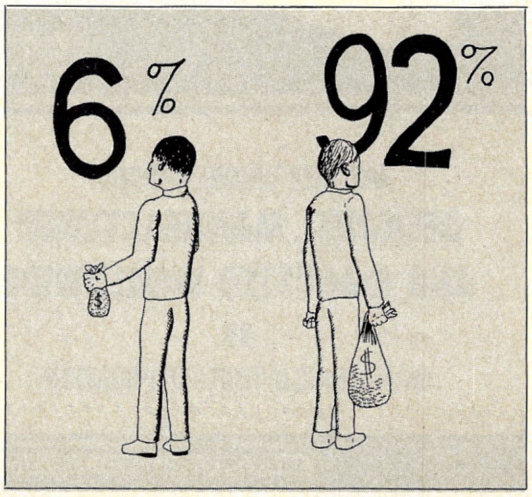
A boy and his sister visited a farm where they saw a pen filled with pigs and chickens. When they returned home, the boy observed that there were 18 animals in all, and his sister reported that she had counted a total of 50 legs. How many pigs were there in the pen?

Typically the response to this problem is as follows: approximately 40% of the students don't know how to begin because they haven't studied any specific methods for this problem, 25% recognize that the problem could be solved with two equations and two unknowns but have forgotten how to do it, 25% can set up the two equations and get a solution, and 10% quickly try a few numbers and get the answer by trial-and-error in two or three tries. On the other hand, when we give the same problem to elementary-school children who know nothing about two equations and two unknowns and again ask them to solve it using any method they choose, they turn to trial-and-error very naturally and a higher percentage answer it.

What does this all mean? To us it indicates that in the teaching of mathematical and scientific problem-solving in school we overemphasize the memorization of specific devices and techniques for attacking problems, and we underemphasize some very simple and powerful problem solving strategies useful for a variety of problems. In this article on general problem-solving strategies, we would like to turn the tables around and elaborate on the strategy of trial-and-error, which is always available but seldom used to its full potential. It is often frowned upon in school because it is thought to be a lazy approach which requires very little thinking. But with the computer available, trial-and-error takes on a whole new dimension which we will only begin to explore here.

taxes problem

The strategy of trial-and-error can be used in a number of different ways, which we will illustrate with the following problem about income taxes.



*University of Wisconsin-Parkside, Kenosha, Wisconsin 53141.

TAXES IN TAXES???

Naturally, many people believe that rich people should pay more taxes than poor people, since the wealthier ones have more money. But sometimes this policy is carried to extremes. In one place I recently heard of, the tax rate was made the same as the number of thousands of dollars a person earns. For example, if a person earns $6,000, then his tax rate is 6% of that. But if a person earns $92,000, then his taxes are a whopping 92% of that.

What income between $1 and $100,000 would leave you the most money after taxes?

One way to solve this problem is *random* trial-and-error. As the term implies, random trial-and-error consists of arbitrarily choosing a series of values (gross salaries in this case) calculating results (net salaries), and then testing to see which one yields the highest value. This method takes little thought and only produces a solution if one happens to pick the correct value or values. In the case of the taxes problem the method would be extremely inefficient because each result must be compared to all previous ones to see if it is larger. It is this type of an approach to problem-solving that has given trial-and-error a bad name.

An improvement over random trial-and-error that requires additional thought and substantially improves its utility is *systematic* trial-and-error. Here a rule is devised to make certain all the reasonable alternatives will be systematically considered and evaluated until a problem is solved or shown to be unsolvable. With the tax problem this might consist of beginning with $1,000 and trying successive values in increments of $1,000 until a solution is reached. The result is that the net pay would continue to increase up to a gross income of $50,000 and then begin to decrease. This result implies, of course, that the optimum income is $50,000. While this method may be tedious and time-consuming, it will usually produce a solution, and therefore is a substantial improvement over random trial-and-error.

A further refinement of trial-and-error, and one that makes it much more respectable as a general problem-solving strategy, is *guided* trial-and-error. The key to its success lies in the fact that the results from each trial are used to guide the choice of a value for the next trial that will produce a result closer to the correct solution. This process is continued until the solution is finally attained, and is usually much more efficient than systematic trial-and-error. As an example of guided trial-and-error, let us return to the taxes problem. As a starting point, we might try both values suggested in the problem ($6,000 and $92,000). Because the results show that $92,000 yields a higher income than $6,000, it seems reasonable to choose a value higher than $92,000. As it turns out, however, values higher than $92,000 provide less net income than does $92,000. Therefore, it is logical to choose values less than $92,000 in large increments ($10,000) as long as they continue to result in larger net incomes than values chosen previously. Because the correct answer is $50,000, values closer to $50,000 will produce larger and larger net incomes, and values less than $50,000 will produce decreasing net incomes. Thus, the correct value can be determined quite efficiently. As mentioned earlier, the key to success is to carefully examine the result from each trial to guide the selection of values for the next trial in a way that will guarantee a movement closer to the correct solution.

systematic trial-and-error

Now we turn to the computer and write a program to find the solution using systematic trial-and-error. If X represents the gross income then .01X is the tax rate and f(X) = (1-.01X)X is the net pay. Program 1STMAX uses a systematic procedure to find the value of X for which the net pay achieves a maximum value on the interval 0 to 100. The system is based on the following principle: Let X=0 be the starting point and increment to the right by I=10, comparing f(X) with f(X+I). If f(X)< f(X+I) then move up one step (X=X+I) and compare f(X) and f(X+I) again. Continue this procedure until f(X)>f(X+I). Now move back one step (X=X-I), reduce the step size by a factor of 2 and continue as before. As soon as the step size falls below the specified level of accuracy D (I<D), print out the value of X that corresponds to the first maximum value of f(X). For functions that have more than one relative maximum, the systematic procedure used in 1STMAX can be easily extended to find all relative maximum and minimum points for a given function on a specified interval. Can you do it?

1STMAX PROGRAM

```
LIST
1STMAX

10  PRINT "***THIS PROGRAM FINDS THE X VALUE WHERE THE FUNCTION X-.01*X*X"
20  PRINT "***IS A MAXIMUM ON THE INTERVAL 0 TO 100 BY SYSTEMATIC TRIAL"
30  PRINT "***AND ERROR."
40  PRINT
50  PRINT "INPUT THE DESIRED DEGREE OF ACCURACY.";
60  INPUT D
70  DEF FNF(X)=X-.01*X*X
80  PRINT
90  X=0
100  I=10
110  PRINT " I","X AT MAX"
130  Y1=FNF(X)
140  Y2=FNF(X+I)
150  IF Y1>Y2 THEN 180
160  X=X+I
170  GOTO 130
180  PRINT I,X
190  IF I<D THEN 230
200  X=X-I
210  I=I/2
220  GOTO 130
230  PRINT
240  PRINT "THE X VALUE WHERE X-.01*X*X IS MAXIMUM IS";X
250  END
```

SAMPLE RUN

```
RUN
1STMAX

***THIS PROGRAM FINDS THE X VALUE WHERE THE FUNCTION X-.01*X*X
***IS A MAXIMUM ON THE INTERVAL 0 TO 100 BY SYSTEMATIC TRIAL
***AND ERROR.

INPUT THE DESIRED DEGREE OF ACCURACY.?.1

 I              X AT MAX
 10              50
 5               50
 2.5             50
 1.25            50
 .625            50
 .3125           50
 .15625          50
 .078125         50

THE X VALUE WHERE X-.01*X*X IS MAXIMUM IS 50
```

x•log(x)=100

Although the taxes problem can be solved analytically using calculus or even more easily with the properties of quadratic functions, it is more likely that the majority of non-textbook problems one encounters will not have a nice closed-form solution. One such example is the following: *Find a value of X such that X•log(X) = 100.* (We assume here that log(X) is the natural logarithm.)

Systematic trial-and-error, which is frequently used to search for solutions with a computer, could be applied to this problem in much the same way it was applied to the taxes problem. But for variety, we will use a different type of systematic trial-and-error.

Clearly, the solution to X• log(X) = 100 lies somewhere between 1 and 100 since 1 • log(1)=0, 100 • log(100) > 400, and X•log (X) increases with increasing values of X. If we let X_l be the guess that is low (X_l •log (X) < 100) and let X_h be the guess that is high (X_h • log (X_h) > 100) then we can use a systematic procedure which generates new trials by dividing the search area in half at each step as follows: Let the new trial X_n be the average of the last two trials X_n = (X_l+ X_h) /2 and then test X_n to see whether it is high, low, or within the accuracy desired. If X_n is high (X_n•log(X_n) > 100) then replace the last high guess with X_n (X_h = X_n) or if X_n is low (X_n • log(X_n) 100) replace the last low guess with X_n (X_l = X_n) and repeat the process of taking averages. Since the distance between X_l and X_h is cut in half with each new trial, X_l and X_h will both approach the desired solution within any pre-set degree of accuracy given a sufficient number of iterations. Program SLICE solves X•log(X) = 100 for X using this method. The sample run following the program lists the upper and lower bounds at each halving of the search area to illustrate the approach. When this printout is suppressed the answer is computed immediately.

SLICE PROGRAM

```
SLICE

10  PRINT "*****THIS PROGRAM SOLVES X*LOG(X)=100 FOR X"
20  PRINT "*****BY USING SYSTEMATIC TRIAL AND ERROR."
25  PRINT
30  PRINT "INPUT A LOWER AND UPPER GUESS AND THE DESIRED ACCURACY";
40  INPUT X1,X2,D
50  N=1
60  Y1=X1*LOG(X1)
70  Y2=X2*LOG(X2)
80  PRINT
90  PRINT " N","X-LOWER","X-UPPER"
100  PRINT N,X1,X2
110  X3=(X1+X2)/2
120  Y3=X3*LOG(X3)
130  IF ABS(Y3-100) <= D THEN 210
140  IF Y1<100 AND Y3>100 THEN 180
150  X1=X3
160  N=N+1
170  GOTO 100
180  X2=X3
190  N=N+1
200  GOTO 100
210  PRINT LIN(1)"THE ANSWER IS "X3
220  END
```

SAMPLE RUN

```
*****THIS PROGRAM SOLVES X*LOG(X)=100 FOR X
*****BY USING SYSTEMATIC TRIAL AND ERROR.

INPUT A LOWER AND UPPER GUESS AND THE DESIRED ACCURACY?1,100,.001

 N            X-LOWER       X-UPPER
 1             1             100
 2             1             50.5
 3             25.75         50.5
 4             25.75         38.125
 5             25.75         31.9375
 6             28.8437       31.9375
 7             28.8437       30.3906
 8             28.8437       29.6172
 9             29.2305       29.6172
 10            29.4238       29.6172
 11            29.5205       29.6172
 12            29.5205       29.5688
 13            29.5205       29.5447
 14            29.5326       29.5447
 15            29.5326       29.5386
 16            29.5356       29.5386
 17            29.5356       29.5371
 18            29.5364       29.5371

THE ANSWER IS  29.5367
```

guided trial-and-error

The systematic trial-and-error algorithm for solving X• log(X) = 100 given above does not take into account all the information available aftereach trial. For example, this method takes the same amount of time to reach a solution whether the first guess is close to the solution already or not and is independent of the problem being solved. On the other hand, guided trial-and-error uses more of the information available from each trial (such as how close a particular trial is to a solution) to make a more educated next trial. This technique is used in program GUIDE to solve the problem X•log(X) = 100 and is based on the following principle: Let X_n be a given trial and Y_n = X_n• log(X_n) be the corresponding value of the function. If Y_n is too large (Y_n > 100) then the exact trial is decreased by the factor 100/Y_n which is less than one. If Y_n is too small (Y_n < 100) then the next guess is increased by the factor 100/Y_n which is greater than one. Thus the new trial is guided by the outcome of the previous trial as follows:

$$X_{n+1} = X_n * 100/Y_n.$$

Notice that this algorithm has the following important properties: Trials that are far from the correct value are changed by a bigger factor than those that are close. All trials oscillate above and below the desired solution. When Y_n reaches 100, all subsequent trials remain the same.

GUIDE PROGRAM

```
GUIDE

10  PRINT "*****THIS PROGRAM SOLVES X*LOG(X)=100 FOR X"
20  PRINT "*****BY USING GUIDED TRIAL AND ERROR."
30  PRINT
40  PRINT "INPUT AN INITIAL GUESS AND THE DESIRED ACCURACY ";
50  INPUT X,D
60  PRINT
70  PRINT " n"," Xn"
80  N=1
90  Y=X*LOG(X)
100  PRINT N,X
110  IF ABS(Y-100)<D THEN 150
120  X=X*100/Y
130  N=N+1
140  GOTO 90
150  PRINT LIN(1)"THE ANSWER IS"X
160  END
```

SAMPLE RUN

```
RUN
GUIDE

*****THIS PROGRAM SOLVES X*LOG(X)=100 FOR X
*****BY USING GUIDED TRIAL AND ERROR.

INPUT AN INITIAL GUESS AND THE DESIRED ACCURACY ?100,.001

 n            Xn
 1            100
 2            21.7147
 3            32.4887
 4            28.7283
 5            29.7807
 6            29.465
 7            29.5578
 8            29.5303
 9            29.5384
 10           29.536
 11           29.5368

THE ANSWER IS 29.5368

DONE
```

newton's method

For completeness, we conclude our search for solutions to the X•log(X) = 100 problem with Newton's method which is a form of guided trial-and-error. This method is fully explained in almost every calculus book so we will not repeat it here. In comparison with the other algorithms given above, Newton's method is a very efficient approach as shown in the sample run. However, there is a price to pay for this efficiency — a knowledge of calculus and the ability to remember the method — which may not be worth the time and effort in many cases. For example, even though Newton's method converges to the solution over four times faster than the systematic trial and error method of program SLICE, the difference in the response time at the terminal is unnoticeable. Besides, there are some problems where Newton's method will fail if one makes an unlucky first guess, whereas the systematic trial-and-error technique never fails.

NEWTON PROGRAM

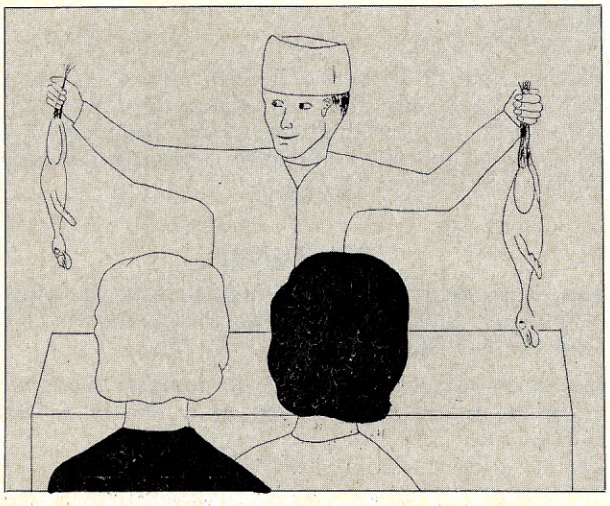
```
NEWTON

10  PRINT "*****THIS PROGRAM SOLVES X*LOG(X)=100 FOR X"
20  PRINT "*****BY USING NEWTONS METHOD FOR GUIDED TRIAL AND ERROR."
30  PRINT
40  PRINT "INPUT AN INITIAL GUESS AND THE DESIRED ACCURACY ";
50  INPUT X,D
60  N=1
70  PRINT
80  PRINT " n"," Xn"
90  PRINT N,X
100  X1=X-(X*LOG(X)-100)/(LOG(X)+1)
110  IF ABS(X-X1)<D THEN 150
120  X=X1
130  N=N+1
140  GOTO 90
150  PRINT LIN(1)"THE SOLUTION IS "X1
160  END
```

SAMPLE RUN

```
RUN
NEWTON

*****THIS PROGRAM SOLVES X*LOG(X)=100 FOR X
*****BY USING NEWTONS METHOD FOR GUIDED TRIAL AND ERROR.

INPUT AN INITIAL GUESS AND THE DESIRED ACCURACY ?100,.001

 n              Xn
 1              100
 2              35.6813
 3              29.6595
 4              29.5367

THE SOLUTION IS  29.5366

DONE
```

turkey puzzle

A completely different application of guided trial-and-error is illustrated in the solution to the *Turkey Puzzle*.



TURKEY PUZZLE

Two people, Jane and Mary, went to the butcher shop to buy turkeys for Thanksgiving dinner. Since Mary had a larger family than Jane, she wanted a larger turkey. The butcher just happend to have two turkeys left, a small one and a large one. "Together these two turkeys weigh twenty pounds," he said. "The little one sells for two cents a pound more than the large one." Jane purchased the little one for 82¢ and Mary paid $2.96 for the big turkey. How much did each gobbler weigh?

Story problems like this one are the bane of most beginning algebra students. They ask "Where do I begin? What method do I use? Did I set it up right?" Frequently, students are primarily concerned about setting up the machinery for a problem so the answer will drop out like an egg into a basket and forget the most fundamental property of any solution: *A solution is an answer that works!* Of course, the Turkey problem can be solved using algebra and the reader may want to try solving it this way. But we will eschew any algebraic devices for attacking this problem to illustrate the power of guided trial and error.

The easiest way to begin the Turkey problem is to try a few numbers. Let's assume, as a first guess, that the small turkey weighs 8 lbs. What implications does this have, given the condition stated in the problem? First, the big turkey must weigh 20 - 8 or 12 lbs. Next, since the big turkey cost $2.96, the price per lb is $2.96/12, about $.25. The small bird cost 2 cents more per lb so its price is $.27. But $.82 was spent for the small bird which means it must have weighed .82/.27, about 3 lbs.

We have come full circle and our results are conflicting. We started out with a small bird weighing 8 lbs and ended up with the same bird weighing 3 lbs. If the first guess and the outcome had agreed for the weight of the small bird, the system would have been consistent and we would have had a solution simply because it worked.

The process of guided trial-and-error is based on the idea of adjusting the next trial depending upon the results of the previous trial. It is a feedback control system similar to the control of a guided missile to its target. From the first trial of 8 lbs the feedback told us we were too high so our next guess should be smaller. In 3 or 4 trials anyone should be able to narrow in on the target of 4 lbs for the small turkey and 16 for the large one.

Would it be possible to set up an automatic trial-and-error or feedback system that is simple enough to be programmed by anyone familiar with the BASIC language and solve the turkey puzzle with only an initial guess? The answer is yes, and program TURKEY is one example of how to do it. The feedback systems of this program follows very closely the first discussion of the Turkey problem given above. If we let L be the weight of the small turkey, B the weight of the large turkey, P the price per lb for the large turkey, then the conditions in the problem can be summarized as follows:

1. B = 20 - L
2. P = 296/B
3. L = 82/(P+2)

Begin with an initial guess of L1 for the weight of the small turkey L and follow through the consequences of this guess in 1, 2, and 3 above. If L1 and L agree, the system is consistent and we have a solution. If not, then use the outcome L in 3 as the next trial in 1 and loop back through the system. This algorithm was used in program TURKEY, and within 7 iterations of the algorithm the system has narrowed in on a solution as shown in the sample run. The program terminates when the difference between successive trials falls below .001.

TURKEY PROGRAM

```
TURKEY

10  PRINT "SOLUTION TO THE TURKEY PUZZLE BY GUIDED TRIAL AND ERROR"
20  PRINT "MAKE ANY GUESS FOR THE WEIGHT OF THE SMALL TURKEY";
30  INPUT L1
40  PRINT
50  PRINT " SMALL"," BIG"
60  L=L1
70  B=20-L
80  P=296/B
90  L=82/(P+2)
100  IF ABS(L-L1)<.001 THEN 140
110  PRINT L1,B
120  L1=L
130  GOTO 70
140  PRINT
150  PRINT  USING 160;L+.001,P+2.001
160  IMAGE "THE SMALL TURKEY WEIGHS",2D.2D," LBS AND THE PRICE PER LB IS ",2D.2D
170  PRINT  USING 180;B+.001,P+.001
180  IMAGE "THE BIG TURKEY WEIGHS ",2D.2D," LBS AND THE PRICE PER LB IS ",2D.2D
190  END
```

SAMPLE RUN

```
RUN
TURKEY

SOLUTION TO THE TURKEY PUZZLE BY GUIDED TRIAL AND ERROR
MAKE ANY GUESS FOR THE WEIGHT OF THE SMALL TURKEY?8

 SMALL            BIG
 8                12
 3.075            16.925
 4.20752          15.7925
 3.95312          16.0469
 4.01057          15.9894
 3.99761          16.0024

THE SMALL TURKEY WEIGHS 4.00 LBS AND THE PRICE PER LB IS 20.50
THE BIG TURKEY WEIGHS 16.00 LBS AND THE PRICE PER LB IS 18.50

DONE
```

conclusion

The development of digital computers has revolutionized the methods available for solving problems. In many areas of modern science, computer-oriented numerical methods have been developed to solve problems that have been impossible to solve analytically. The diversity of fields being affected includes planetary astrodynamics, wave diffraction, weather prediction, thermodynamics, electrostatics and gravitational potential, molecular interaction, quantum theory, and relativistic collapse (Greenspan, 1974).

The guided trial-and-error methods discussed above are but a small sample of the growing number of computer techniques being used to find solutions or approximate solutions to given mathematical problems. The immense power of the digital computer to perform arithmetical operations with exceptional speed has added a quantum jump in the number of techniques now available to solve problems.

postscript

The iteration technique used in the TURKEY program will converge to the solution, given any initial guess. This of course is the desirable outcome, but it is only one of three typical outcomes. An iteration can diverge away from any solution, getting increasingly larger with each iteration or getting locked into an endless loop that oscillates between two numbers that are not solutions. It is remarkable that the outcome that does occur depends only upon the way the conditions are expressed. For example, if you try writing a program to solve the Turkey puzzle using the following equivalent set of conditions:

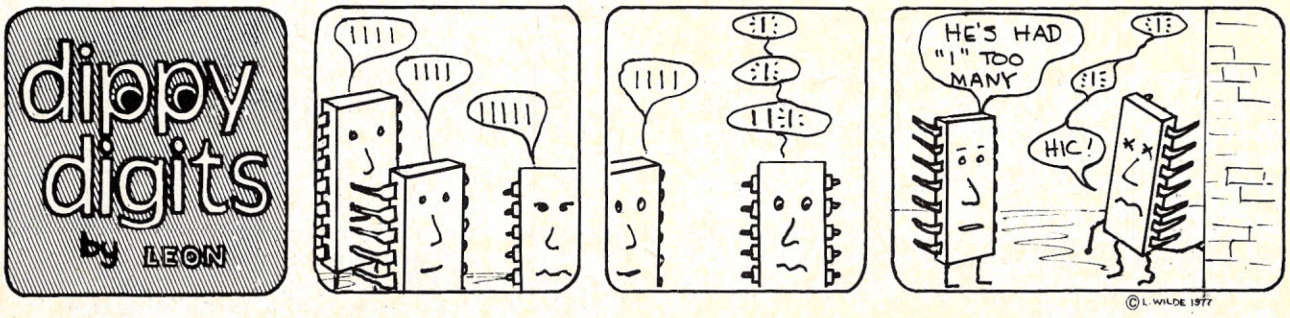
1. L = 20 - B
2. P = 82/L - 2
3. B = 296/P

the sequence of iterations diverge and no solution is reached. Try it! The above equations were obtained by inverting the original three equations. Therefore, if you use a set of equations that give a diverging outcome, simply invert them and try again.

REFERENCES

Piele, D. T., and Wood, L. E. "Thinking Strategies with the Computer: Inference", *Creative Computing Magazine,* Vol. 3, No. 2, Mar-Apr., 1977, p. 96-99.

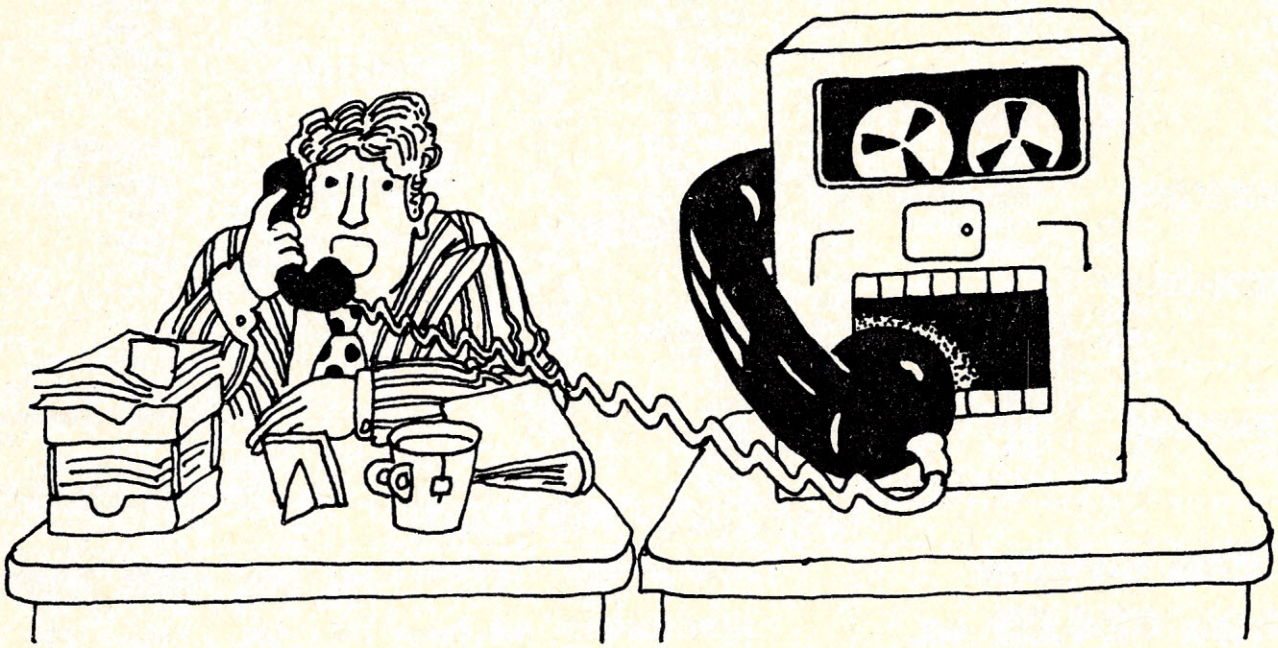
Greenspan, Donald. *Discrete Numerical Methods in Physics.* Academic Press, Inc., New York, N.Y. 10003, 1974. ■



"One can easily conceive of a point in time somewhere in the 1980's where 80% of what used to flow via first-class mail is handled by computerized conferencing. . . ."

Computerized Conferencing: Present and Future

By MURRAY TUROFF
Professor of Computer Science, New Jersey Institute of Technology, Newark



TO DESCRIBE a communication process is, in essence, to describe a human experience. The author is not completely satisfied that this article, as a "one-way" mode of communication, is adequate to convince the reader that the introduction of a computer as a mechanism for fostering human communications creates a new and unique communication media with characteristics that impact on the attitudes and psychology of those engaged in the process. A much simpler mechanism for accomplishing this understanding, were it possible, would be to invite the reader to sit at a computer terminal and actually live the experience. However, that being impossible, let us proceed under the limitations imposed by this approach.

Characteristics of Communicating via Computer

To create a perspective for comparison, let us imagine that two of us, along with others, are engaged in a verbal discussion. This might be occurring in a face-to-face mode in the same room, or via a telephone or television conference mode. The verbal mode of communication has certain rules and procedures dictated by the nature of verbal exchanges. First and foremost is that only one person may speak at a time if the conversation is not to become confusing to the group as a whole. If you have something to say which pertains to the current speaker's remarks, you must normally wait until he is finished, and even then you may not have the opportunity to interject your comments if someone else gains the floor ahead of you. In fact, the psychology of verbal communication is such that when you have obtained the opportunity to speak, the topic may have shifted to the extent that your planned remarks are out of place or too late. Therefore, the verbal exchange represents a "synchronous" form of communication where individual participation is not only sequential in nature, but also under the control of the group and whatever explicit or implicit rules of order they have established. In this atmosphere, one

may either listen or talk when *allowed* to. If this same group communication were to take place on the computer, one might either be "talking" by typing his remarks, or "listening" by reading the remarks of others as they were printed on the terminal. However, since the computer stores the remarks until each individual indicates he is ready to receive more of the conversation, the individual is now able to control his rate of participation in the conversation. He may choose whether he wishes to listen or to talk at any given instant. If he listens, he may, of course, decide to read at a rate which is desirable to him for the possible purposes of skimming, pondering, reflecting, or even ignoring what is being said. If he talks, his remarks will be added on the end of the list containing the conversation and will be received by others when they have reached that point in the conversation. The group acts in an asynchronous manner, and all could be talking or listening at the same time or any intermediate combination. Because the individual controls his communication rate and time, the psychologists would term this manner of communicating a "self-activating" process.

In practice, a discussion is actually a number of separate discussion threads becoming interleaved. As a result, there is not the same pressure to restrict the discussion to a sequential flow with respect to the specific topic of the moment. Therefore, individuals who wish to think about what they say on a particular matter may wait for a time before making their remarks. That some of the others in the conversation may have moved on to another topic does not detract from the ultimate impact of the comments. Furthermore, since the computer assigns a unique sequence number to each message (in the order of occurrence) and labels it with author, date, and time, a later message referring to an earlier one need only begin with "Ref. mes. #101." This is in sharp contrast to a verbal discussion, where a typical comment referring back usually begins: "In regard to what John was saying awhile back about such and such. . . ." While there is a learning curve for effective use of this communication mode, a group communicating in this manner becomes accustomed to this oscillating form of communication after a few hours of practice. Individuals quickly learn to refer back in their remarks to a specific earlier comment they are discussing. In addition, the written form encourages compactness and precision of expression. Furthermore, the sorting capability of the computer can be used to regroup the discussion into its separate threads by utilizing unique identifiers—such as the sequence numbers or key words used in the discussion itself—to define the particular topic. Indeed, individuals communicating through such a system tend to develop a feeling of equality with the other group members. The resulting group atmosphere is very different from a committee meeting where some one individual usually takes control (even if only tacitly) for the purpose of sequencing the discussion. This does not say, however, that the group itself does not wield impact on the individual. If an individual is not making sensible or pertinent comments, or if his remarks are verbose and drawn out, he quickly discovers that no one else is reflecting or commenting back on what he has said. In fact, he quickly begins to wonder if anyone is bothering to read his contributions. This mode of communication has its own unique set of psychological factors and pressures.

Any group which can be inter-connected with a conference telephone can now interact via computerized conferencing. But there is no limit on number of people or the time of the call.

In a face-to-face discussion, it is impolite for a member of the discussion to whisper to his neighbor or to pass notes to another member of the discussion. It is certainly not done as often as individuals might desire to do so. In the computerized version of conferencing, any individual may write a private message to any other individual, and the rest of the conference members will never even know it has occurred. These messages may also be copied to any sub-group. Therefore, two individuals may arrive privately at a joint view on some issue or on the remarks of another individual, or some individuals may make lunch plans or a couple may carry on a flirtation. This universal whispering capability is available for any use an individual in the conference desires to make of it. For certain applications, this ability to carry out timely sub-group negotiations could lead to more rapid resolution of some particular issue. These private messages are destroyed from the conference file as soon as they have been delivered.

The design of a sub-group negotiation capability can be carried further. Designs can be tailored specifically for groups having severe differences of opinion or engaged in intense negotiation. Such tailored communication systems could be designed either to foster compromise or to strengthen the opposing views, depending on the objective of the discussion.

Any group which can be inter-connected with a conference telephone can now interact via computerized conferencing, provided they can obtain access to computer terminals. Therefore, when the issue to be discussed by the group is time-urgent and either the available time or funds is not sufficient for necessary conference travel, computerized conferencing offers distinct advantages over a telephone conference mode. In addition, computerized conferencing also offers the significant secondary effect of making it possible to have 30 or more people engaging in a useful give-and-take discussion. A conference telephone call is cumbersome with more than five people, and often degenerates to a mechanism useful only for allowing a group to be quickly informed of one person's views or orders.

While the utility of computerized conferencing for geographically dispersed groups is somewhat obvious, we have not touched on a dimension of far greater impact—time dispersion. Since the conference dialogue is stored, it is not necessary for individuals involved

to be on the computer terminals at the same time. A person may go to the terminal at a time that is convenient to him, and the computer will tell him how much of the conversation each member has received and how recently each has been active. He may then receive any messages he has not previously seen, make his additional comments, and sign off. The next person to sign on will also find these additional comments, and anything else he has not seen previously. The individuals engaged in this random mode of conferencing may now control the use of their time to a much greater degree than is possible when a group must simultaneously meet for a discussion.

The computer, therefore, not only allows a person to control his rate of interaction when he is participating in the conversation, but even when he wants to start or stop engaging and to trade that off with other demands of his time. He is no longer a "slave" to the demand of having a single time for communication which corresponds with every other individual in the group.

One of the most pragmatic applications is straightforward project management. The manager no longer needs to be interrupted by a telephone call everytime one of the 10 to 20 people working for him has a question. Instead, he can turn to the terminal every few hours, at his choosing, and answer these questions at one time. The resulting conference file then becomes a documented history on the project, with all modifications and agreements recorded.

Mechanics of Operation

When a participant in a conference system calls up for a particular session, he will usually request at the outset a list of the current status of the participants to determine when each of them was last on, and how many messages each received out of the message list which makes up the conference. If anyone else is currently interacting in the conference when he gets on, or should someone get on while he is on, he will automatically be informed of who it is. He will then respond to computer requests for his name and security code, which logs him in, after which the computer will begin to list for him all the messages that came into the conference since he was last on. At the completion of that list, he will be asked for any message he wishes to add. When he finishes typing a message, he uses a special symbol—in our case, the "+" sign—to have the message added on to the conference file. Before doing so, however, he has available a number of editing capabilities which allow him to skip around the text of his message for correction of errors. After he sends his message, the computer will list any messages which came in while he was writing his own, and it then returns him to the writing mode for the entry of a new public message.

In this mode, he may also take advantage of a number of commands that are available for special situations: writing a private message; editing an earlier message; retrieving messages edited in a certain time frame; retrieving messages with certain words and phrases in them; retrieving all messages written by one participant; asking the terminal to "go to sleep" for a specified number of minutes, or until a particular person does something, or until anyone does something; writing, but delaying entry of a message until a specified date and time; entering a proposal; voting on a proposal; retrieving the votes on proposals, etc. All these options are usually summarized in a one-page user's guide. One of the difficulties associated with a conference telephone call is that it is tedious to keep establishing who is still on the line—*i.e.,* accidental cutoffs produce no signal—and who desires to talk at a given time. In a face-to-face meeting, this is accomplished visually and produces a smoother flow of conversation. The computerized conferencing system is structured to automatically inform the group when someone joins or leaves the discussion—*i.e.,* who has walked in or out of the meeting room. Also, the status of the group members, with respect to their activity and position in the discussion, may be retrieved at any time. The computerized conference thus provides more of the atmosphere of a face-to-face exchange than a telephone conference call, at least with respect to the visual "signals" of the face-to-face interaction.

When the user has finished his interaction, he signs off. The computer will keep a record of his location in the discussion and pick up at that point when he next gets on—much as one would leave a bookmark in a book. In a sense, one may view the computerized conference as the simultaneous writing of a book, paragraph by paragraph, by a group of authors working together. The casual user or the new user need only to learn how to get on and off, write a message, and send it into the central file. Retrieving comes automatically. The average user learns how to do these operations in about 10 minutes, provided that he is using one of the better-designed interactive terminals. People with no computer experience at all usually achieve quite adequate facility in using the system and most of its capabilities in a half-hour to an hour. For a group to get used to communicating proficiently about an involved issue via this method involves a few additional hours of practice. Once the conversation is in the computer, a large number of options can be offered the user. Some examples are: conditional messages which only go to the group when a particular person writes a message which contains a certain word or phrase; the ability to send a message on an anonymous basis; automatic spelling correction when dealing with children or people learning the language; automatic association capability so each individual can build his own links or associative structure of the conversation; messages with a finite life time; delayed messages which only go to the group on some specific future date and/or time; and messages with vote scales attached for group voting on items, such as proposals.

The straightforward discussion structure can be modified for particular applications. For example, two or more copies of the discussion can be set up to represent different languages, and *human* translators can interface with the system to translate comments from one language to the other. Certain communication protocols can be built in, whereby a writer of an English message could request to see the French copy of his message

Topic-oriented "blind-dating" conferences may be maintained to help people of similar interests to find one another.

before it is transferred to the French copy of the discussion; or he could request to have a second translator translate the French back to English so that he can compare it with the original before approving it. If there is a significant difference, he can hold a private sub-conference with the two translators via the system.

Potential Applications

The following potential applications for computerized conference systems exist:

- A group of salesmen involved in marketing a line of technical equipment such as computer peripherals can maintain a continuous conference for the purpose of comparing responses to customer questions and for analyzing competitive products.
- Several medical doctors representing differing specialties in a local area can maintain a computerized consulting and referral network.
- Technical librarians in a group of non-competitive companies may set up a document exchange program and jointly plan complementary acquisitions.
- A decision-maker can direct his subordinate managers to discuss anonymously a set of competitive alternatives for budget allocation among some of the divisions they represent.
- Automobile service managers for a given manufacturer can maintain a conference concerning the merits and performance of test and maintenance equipment, or concerning unusual servicing problems.
- The manager of a decentralized development effort may use computerized conferencing to maintain with his group the status of his various projects and their current specifications.
- A committee which meets regularly only once a month can use computerized conferencing to maintain continuous contact and to arrive at the agenda for its face-to-face sessions.
- Use by the deaf as a conference phone type system.
- Use by home-bound handicapped to remotely participate in educational courses.
- An author of a technical paper can discuss it with a group of referees via an anonymous computerized conference.
- Members of legislatures can caucus at will with a computerized conference.
- Policy-makers can obtain quick response on the pros and cons of critical issues from consultants scattered around the country.
- A computerized conference may be used for negotiations among non-anonymous labor and management people and several anonymous arbitrators.
- Teenagers can maintain an anonymous conference on dating customs and problems.
- A college student taking a course in anthropology may participate in a role-playing conference simulating a primitive society.
- A housewife may join a local conference discussing current affairs.
- A stamp or coin collector may join a regional conference for the purpose of arranging trades.
- A family can join in a simulation of planning for the local area as an evening's form of entertainment.
- Students can conduct an anonymous conference with their teachers to discuss various problems and subjects in a course.
- Topic-oriented "blind-dating" conferences may be maintained to help people of similar interest to find one another.
- Delphi designers can maintain a continuous conference to compare notes on current work in the field (a revitalization of the effectiveness of "hidden" research communities).
- Multi-language conferences may be called with human translators on-line to do real-time translation—certainly more feasible than computerized translation.

These potential (in some cases, real) applications serve to introduce the many possibilities. Once computer conferencing systems are recognized as an alternative form of communication, it becomes obvious that almost any endeavor involving human communication can be handled by computerized conference systems, sometimes with surprising side effects. However, it is the ability of these systems to handle large groups, provide hard copy, preserve anonymity, and allow participants to control their time and rate of interaction that makes the concept attractive for many applications which would be difficult, ineffective, or inefficient with conventional alternatives.

Cost Considerations and the Future

The biggest bottleneck to widespread implementation is not the central computer, but the cost and availability of terminals and communications. Terminals will reach fairly low cost, comparable to color television sets, in the late 1970's. For communications, we must ultimately have rate scales based upon amount of data transmitted, as opposed to time charges, made available to the general public. Right now, only organizations able to finance participation in digital data networks or private line systems are capable of enjoying this type of service over a geographically wide area.

One can easily conceive of a point in time somewhere in the late 1980's where 80% of what used to flow via first-class mail is handled by computerized conferencing and other store-and-forward digital message systems. Moreover, the availability of these systems will have a tremendous impact on transportation/communication substitutability. Some day, we should reach the point where the citizen can have the option of phoning from his home for a catalog of on-going conferences and then dialing and joining a particular conference on a topic of interest to him—stamp trading, a new book, a group therapy session, marital problems, etc. When this happens, people will have an efficient method for finding others of similiar interests in the society. That type of capability will, in its own way, change and influence the very structure of the society itself. At the very least, it would offer an active form of entertainment, as opposed to the passive nature of broadcast TV. ■

The Computer Conference:

An Altered State of Communication?

Using ordinary telephone lines, people can now join an invisible network and attend a conference that runs continuously, 24 hours a day, for as long as the participants want. After analyzing some 5,000 hours of such computer conferences, researchers at the Institute for the Future in California believe that this unique medium can create an altered communication state. By enabling people to escape the normal bounds of time and space, computers may thus provide an opportunity to create and explore new patterns of human expression.

by Jacques Vallee, Robert Johansen, and Kathleen Spangler

Most of us communicate intuitively. We talk every day without any thought of the contracting muscles of our vocal cords, or the semantic intricacies of our language, or the subtle visual clues of our individual "body language". The social demand for immediate communication in a face-to-face situation sometimes pressures us into indifferent or inaccurate responses. Our limited ability to remember words often causes us to lose much of the conversation. Limitations of time and place restrict us even further; despite the modern invention of the telephone, a Californian still has to consider whether the Londoner he wants to call will be asleep when he makes the call. But today computers have opened the door to a unique form of communication that may create new patterns of human expression.

Most of the intuitions — and the limitations — of traditional human communication do not apply to computer communications. Using ordinary telephone lines, people can now join an invisible network and attend a conference that runs continuously, 24 hours a day, for as long as the participants want. In computer conferences, time and distance are dissolved and visual cues no longer exist. Each person's "memory" of what has been said is accurate and complete. Everyone may speak at once or listen at leisure. With such features it is not surprising that computer conferences may establish new states of communication in which entirely different patterns of interaction emerge.

A scant 100 or so persons throughout the world now use computerized conferencing on a regular basis. But the time may be fast approaching when far more people will be conferring through computers and we will begin to view the computer as a "natural" means of communication.

THE AUTHORS ARE RESEARCHERS AT THE INSTITUTE FOR THE FUTURE, 2740 SAND HILL ROAD, MENLO PARK, CALIFORNIA 94024.

FORUM, the computerized communication system developed at the Institute for the Future in Menlo Park, California, has enabled its participants to escape the normal bounds of time and space, and researchers who have analyzed some 5,000 hours of computer conferences believe that the system opens up new vistas of human creation and expression.

Unlike face-to-face gatherings, FORUM conferences are characterized by the physical isolation of each participant. Alone with his terminal, each conferee depends on an unseen computer to communicate with his colleagues. All "conversation" is typed on a computer terminal with a standard typewriter keyboard. As a result, accessibility and reliability of terminals, typewriting skills, and writing ability — factors which are never considered in traditional contacts — all influence communication in a computer conference.

While at first it might appear that isolated persons conferring through remote keyboards would be little more than extensions of their machines, this has not proven to be entirely true. One conferee reported that "relationships were established easily, personalities came across, conversations could be established." A computer system such as FORUM reduces the consciousness of distance since it typically costs no more to "talk" across thousands of miles than across the street, but even more striking is the unique "suspended time" of a computerized conference. Participants may enter and leave the discussion at will, without risk of losing touch with the meeting. Pressure to respond to a question immediately is greatly reduced. The participant may take time to consult a library, review his own thinking, and present a well-prepared response. Lack of pressure has not been proven to lead to lack of motivation. Researchers have found that direct questions usually receive prompt replies, and conference growth curves show that the majority of conferences have

Condensed from an article which originally appeared in *The Futurist*, June 1975.

constantly or positively accelerated growth rates — an indication that the momentum of the conference can generate pressure to communicate.

Although computerized discussions are usually more formal and structured than the familiar face-to-face meetings we are accustomed to, this new medium of communication allows not only innovative patterns for communication, but also new opportunities for studying these human interactions. Perhaps as many as 50 researchers in the world are doing work on the social effects of the computer conferencing medium. The theoretical basis for their work is rich, providing two powerful tools for evaluating its social characteristics: an up-to-date machine readable transcript of every computer meeting is always available, and the computer can unobtrusively map interpersonal interactions to reveal patterns of communication among individuals, groups and subgroups.

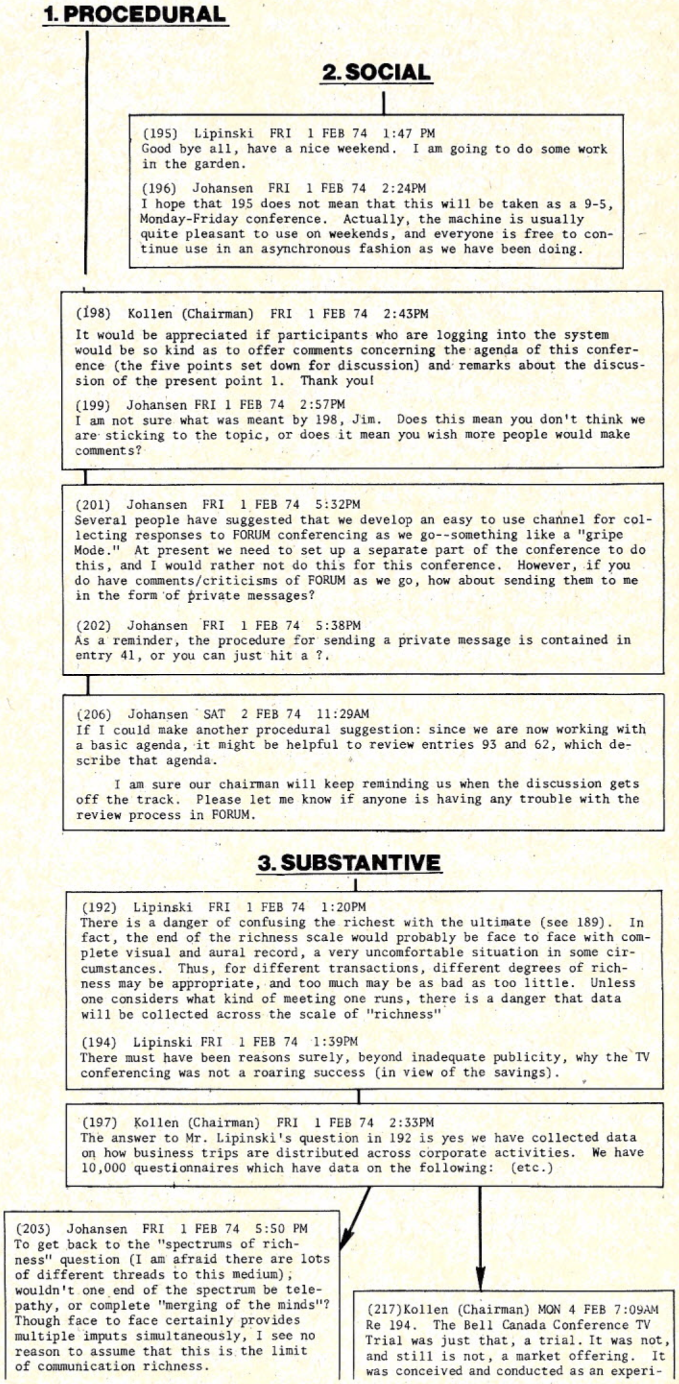
Researchers have discovered that they can identify the roles that different participants play in the development of various topics. Some persons tend to introduce new ideas, others are best at developing them; still others function as synthesizers. The computer itself can map many other dimensions of the interaction that may or may not be evident from the transcript. Comparative participation rates, growth curves, daily activity, and other related indicators create new dimensions for assessing group interaction. Private message statistics may indicate the formation of subgroups, cliques, or coalitions. Individual participation can be traced from one conference to another to judge a person's performance in a certain topic or task. And all of these statistics can be compiled independently of the content of the conference, without violating the privacy of the conference, or the conferee, a task that is just about impossible for the traditional social researcher.

Despite its current limited availability, the new communications medium has universal implications. "Invisible colleges" may develop, introducing and coordinating groups of people who may or may not have been in touch previously. Scholars, businessmen, and government officials would be able to interact outside the normal limits of time and space. Group creativity might be enhanced with everybody at a conference thinking and expressing his thoughts in multiple streams, enhancing our collective abilities to resolve conflicts dealing with crises, or improve decision-making capability. Perhaps computer conferencing will spawn new types of poetry or literature.

On the practical side of the picture, a portable computer terminal hooked to a standard telephone line, would provide an ill or handicapped person with a channel to the outside world. The "coolness" of the medium could also be useful for such activities as psychotherapy, encounter sessions, counseling and discussions of personal values, because in some cases the isolation of the system might be less threatening than a face-to-face group.

Computer based conferencing as a medium of communication is not yet ready for the masses. Although systems such as FORUM can be used by anyone whether he has any knowledge of computers or not, have been developed, they are still quite expensive (about $15 per terminal hour). Heavily loaded computer networks may transmit messages irregularly, resulting in confusion and frustration, since satisfactory communication usually depends on rapid feedback. Total system "crashes" in which the computer completely stops operating can be even more frustrating, and can even prove frightening to someone unfamiliar with the system when the terminal automatically prints out a message like "HOST DEAD." (A British researcher pointed out that a more gentle announcement, such as "HOST PASSED AWAY" might be less traumatic.)

Could computer communications replace much — or all — face-to-face contact? Isaac Asimov, in *The Naked Sun*, and E. M. Forster, in *The Machine Stops*, both wrote nightmarish projections of a future in which electronic communication replaces human contact as we now know it. Researchers today are studying the long-term negative possibilities of computer communications, even as they develop new systems of computer-based conferencing as a supplement — not a replacement — to traditional communication. The potential of computer conferences as a method of communication is still largely unexplored but studies to date lead experts to believe that, all things considered, the new systems offer a new and unique opportunity to enhance the exchange of ideas among people.



The excerpts shown above are from the transcript of an actual computer conference. Participants discussed several topics simultaneously, occasionally dropping one topic "thread" and picking it up later. The communications shown here can be classified as procedural, social, or substantive.

Computer Conferencing: A Personal View

Steve North

In April 1977, Murray Turoff of the New Jersey Institute of Technology invited me to join a new conference administered by NJIT and sponsored by the NSF. I said "Sure, but I'd like it to be participation by any Creative Computing people rather than just me personally." That was agreed to and so we started.

Computer Transceiver Systems of Paramus NJ kindly loaned us an Execuport terminal which gives us good, clean hard-copy output at a reasonable 300-baud speed. To get into the system, one simply dials in to the nearest Telenet port (in every major city in the U.S. and Europe), types a terminal-type code and an access code to connect you to the desired host system. The first couple of times we dialed in, we found the host computer at NJIT was down, but in early May we finally got into some real dialogue. Steve North has been doing most of the "conferencing" to date. His remarks follow. — DHA

I'm not sure whether I should comment on the mundane (specific) or higher (general) aspects of computer conferencing, so you'll get both.

The conferencing system we used, EIES (for Electronic Information Exchange System), is based on a minicomputer. It supports several hundred users though I've never seen more than a dozen users on line at any given moment. The system has been designed to fulfill the needs of both the new user and one who is knowledgeable about the system. Most questions may be answered with a simple "Yes" or "No," or with a number which refers to a choice on a menu. There are many shortcuts for those who don't need to be coached every step of the way along a particular sequence of commands. There is also a buffered-ahead input feature, so that you may input the answers to a whole series of questions at once. For instance, if your next five answers will be Yes, No, 27, 4, and No, you could simply enter "Y,N,27,4,N" to the first question and the computer will not ask for more data until it uses all you input previously. You may also eliminate the printing of menus once you know what you're doing. In short, EIES seems to be of a reasonable design for all sorts of users. As a matter of fact, the users seem to be split about evenly between computer professionals and non-computer people such as social scientists.

There are also several levels at which you may use the system. The simplest method is to use EIES as a message-sending system. You compose a message and send it to another user; the system tells you when there is a message from someone else and prints it. The system also permits users to be grouped into conferences. Users in a conference enter messages into a common area. EIES also has handy commands for reading other users' autobiographies, getting explanations, searching for keywords, etc. In the future EIES will also have a notebook feature (which will let you store text on-line for your own reference), and PILOT, a computer language especially designed for writing computerized dialogs.

What is it like to use a computer conferencing system? It seems to be an ideal means of communication for those who prefer the written word. You compose your message and send it only when you're completely satisfied with it. But you don't have to wait days or weeks for a reply. U.S. Postal Service, watch out! Actually, the threat is that the U.S.P.S. might claim that computer conferencing is competition for first-class mail and attempt to regulate it.

In contrast with a simple computerized message-sending system, computer conferencing is an extremely rich medium. Everyone is in the same conversation, but at the same time you and a smaller group of people may discuss some related or totally unrelated topic. This is possible because the system permits you to direct your messages to an individual, or to a group, or to a public conference. However, the real power of computer conferencing should be the joint decision-making that it makes possible. I don't remember seeing any of that. There are several explanations for that. First of all, most of the messages were either general questions or blueskying. That isn't an area where you can reach a decision. Also, the conference *Creative* was most heavily involved in, Policy and Regulation, was apparently composed mostly of laymen—the real experts had abandoned the conference because the system had been too troublesome to use. The purpose of the conference was to produce part of a paper on computer conferencing, but I have a feeling that the chairman (despite his good intentions) ended up with something not much different than what he would have had if there had been no conference at all.

What is ahead for computer conferencing, or more generally, computer communication? Part of the utility of EIES is derived from the high concentration of users with the same interests. One could safely pick someone at random and strike up a conversation about computers and computer conferencing. Would you pick up a telephone, dial a number at random, and ask "Are you interested in telephones?" Even so, most of the interesting people I met on the system were found by reading their messages or through a reference by a third party, and not through the keyword-searching function of EIES (which is, at best, primitive, and will certainly not be very useful for large-scale systems). Obviously, computer conferencing is special because it uses the power of a computer system. With present technology, a computer conferencing system for the public at large (which doesn't have an interest in conferencing itself) would only be another type of communication for those people who already know each other.

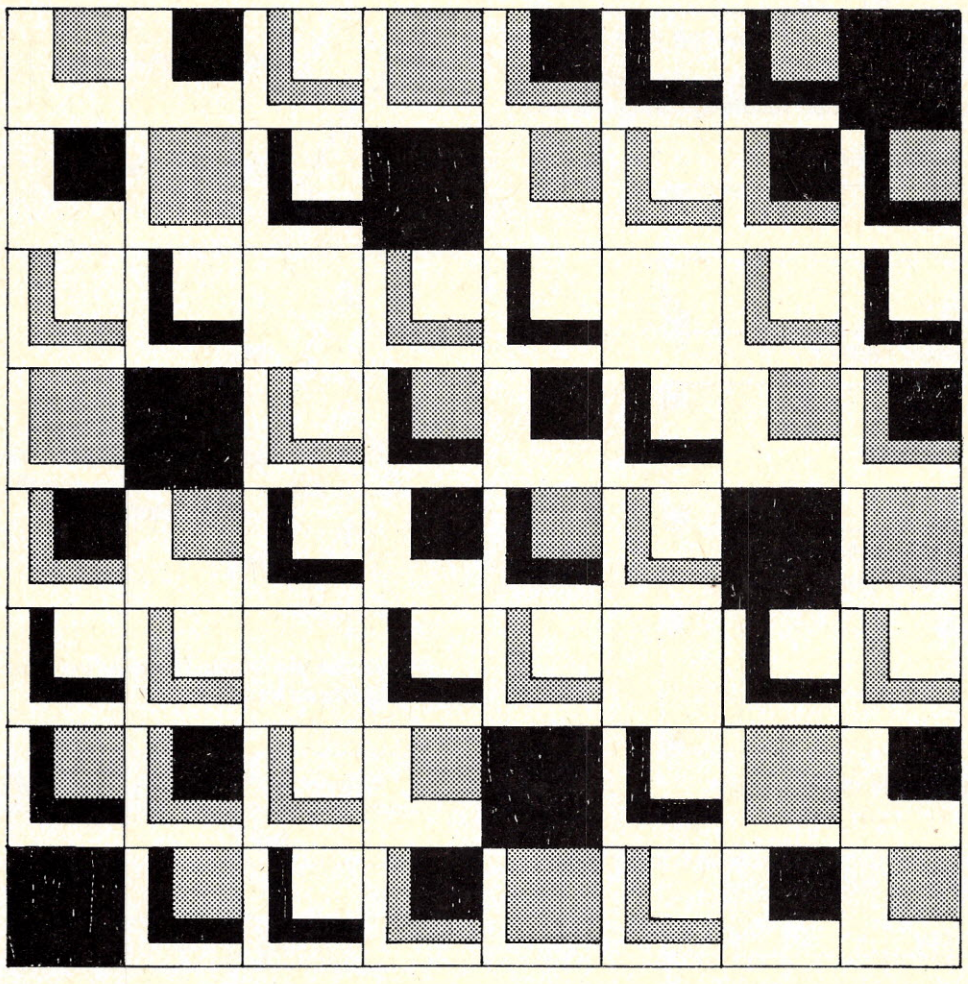
But let's assume that computer conferencing gets beyond all the technological and legal barriers. What will it be like? The "Ultimate" conferencing system might be a part of an integrated communications system for the home and office. After all, why stick with just a keyboard and printer for the man-machine interface? I'm sure that many amateurs could successfully interface their micros with speech-synthesis units to EIES. Graphics are somewhat further down the road.

One last thought. What configuration would be a minimum to set up a usable amateur computer conferencing or electronic-mail system?

- A microcomputer
- A big chunk of memory (40K or more)
- A modem for six users
- An operator's console and interfacing
- At least two floppy-disk units, probably more
- A realtime operating system and software tools for developing the system.

This could be done, I'd estimate, for $10,000 to $15,000. That doesn't include the cost of developing the software to make it all go, which would be substantial. All the text editing would, of course, be done without microcomputers used as smart terminals. ■

ART & MATHEMATICAL STRUCTURES



FLIPS AND SPINS

by R. Chandhok and M. Critchfield

Reprinted from Soloworks Module #2160 — PROJECT SOLO, Dept. of Computer Science, University of Pittsburgh, Pgh., PA 15260

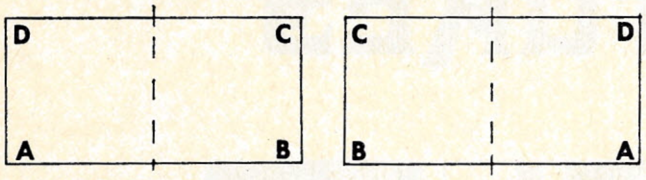
Polygons: The Algebra of Symmetry

There is an important mathematical system that is not based on numbers, but on changing the position of a given polygon.

Imagine a rectangle that is lettered on both sides with the same letter in the same corner. In how many ways can you pick up the rectangle lettered ABCD and then "spin" it or "flip" it so that when you put it down the shape looks the same but the letters are in different positions.

Exercise: Make yourself a paper rectangle like the one above, use it to try out the manipulations discussed below.

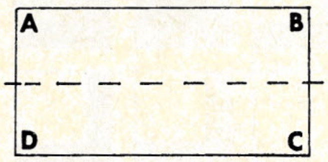
One way is to flip it on its vertical axis of symmetry:



BEFORE FLIP   AFTER FLIP

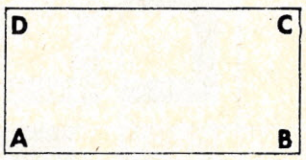
This is the "V" (flip on vertical axis) configuration. The other possible motions (and their resulting configurations) are:

"H" (flip on horizontal axis)



"R" (rotate rectangle 180°)

"I" (itself, a rotation of 0°
or no flip—the identity)



These are the elements of the system:

I, R, V, H

The binary operation of following one motion by another will allow us to create a table like this:

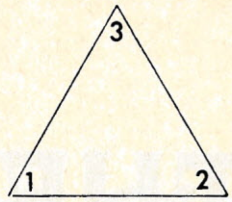
"followed by"	I	R	V	H
I	I		V	H
R	R			V
V		H	F	
H		V		

Exercise: Finish the table by manipulating your rectangle. What are the properties of this table?

Regular Polygons

From here on we will be considering only the regular polygons.

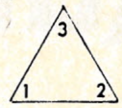
Exercise: Make a paper equilateral triangle. Instead of lettering the corners, number them. Can you do the same motions with it as with the rectangle?



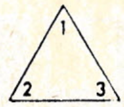
For the sake of simplicity, we will start using the following notations (which can be used for all regular polygons).

R_0 will be a rotation of 0 degrees (replaces I)
R_k will be a rotation of k * (360/n) degrees
where n is the number of sides of the polygon

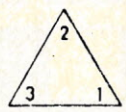
So, for n = 3, a triangle, there are three possible rotations or spins.



$R_0 = 0°$ rotation   0 * (360/3) = 0

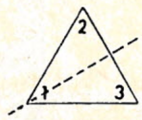


$R_1 = 120°$ rotation   1 * (360/3) = 120

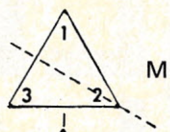


$R_2 = 240°$ rotation   2 * (360/3) = 240

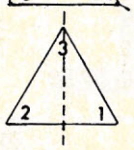
There are also three flips, and a new notation for them.



M_1



M_2 (M stands for "mirror" reflection since flips give the same result as a reflection)



M_3

A table for the 3-sided regular polygon (alias equilateral triangle) would look like this:

"followed by"	R_0	R_1	R_2	M_1	M_2	M_3
R_0						
R_1						
R_2						
M_1						
M_2						
M_3						

Exercise: You guessed it! Use your paper triangle to fill in the above table. Look for its properties.

You can see that it would get quite tedious to try to uncover the properties of all the regular polygons by hand this way.

However, try one more, since there is a slight variation when the number of sides is *even*.

Symmetries of the Square

Using the formula previously given, with n = 4 you can find that the number of rotations of the square are 4—0°, 90°, 180°, and 270°.

Exercise: Make a paper square. How many flips (reflections) does it have? Can you classify them into two kinds? Make a table like the one for the triangle with the spins and flips in the following order:

$$R_0, R_1, R_2, R_3, D_1, D_2, M_1, M_2$$

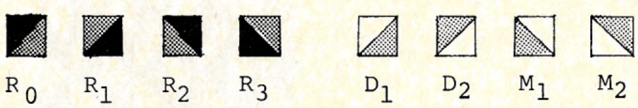
where D stands for "diagonal flip" and M, for a flip that bisects opposite sides.

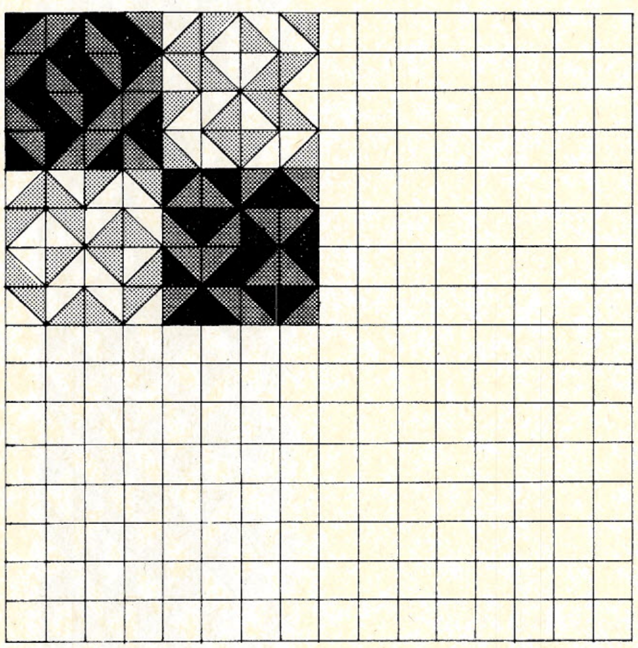
Note: It is possible to write a computer program to produce this, and all other tables from the symmetries of regular polygons. However, it is usually a good idea to go through the construction of tables, by hand, before trying to write a program. An annotated listing and a run of one such program is included at the end of this module.

Exercise: (optional) Make tables for the pentagon, hexagon, octagon, etc.

Designs from Symmetries

Designs based on these tables can be quite surprising and beautiful.





Exercise: (NOT OPTIONAL) Fill in the rest of the above design using colored markers. You decide whether to repeat, reflect, or rotate the original design.

Exercise: (optional) There are other plane figures which have symmetry but are not regular polygons, such as rectangles, parallelograms, trapezoids, and rhombuses. Devise a computer program to produce the table for one of these figures.

ENTER THE COMPUTER

Here's a RUN of the computer program we promised earlier. The program listing is on the following page.

```
? 3

RØ  = 1  2  3
R1  = 2  3  1
R2  = 3  1  2

M1  = 1  3  2
M2  = 3  2  1
M3  = 2  1  3

RØ   R1   R2   M1   M2   M3

R1   R2   RØ   M2   M3   M1

R2   RØ   R1   M3   M1   M2

M1   M3   M2   RØ   R2   R1

M2   M1   M3   R1   RØ   R2

M3   M2   M1   R2   R1   RØ

? 4

RØ  = 1  2  3  4
R1  = 2  3  4  1
R2  = 3  4  1  2
R3  = 4  1  2  3

D1  = 1  4  3  2
D2  = 3  2  1  4
M1  = 2  1  4  3
M2  = 4  3  2  1

RØ   R1   R2   R3   D1   D2   M1   M2

R1   R2   R3   RØ   M2   M1   D1   D2

R2   R3   RØ   R1   D2   D1   M2   M1

R3   RØ   R1   R2   M1   M2   D2   D1

D1   M1   D2   M2   RØ   R2   R1   R3

D2   M2   D1   M1   R2   RØ   R3   R1

M1   D2   M2   D1   R3   R1   RØ   R2

M2   D1   M1   D2   R1   R3   R2   RØ
```

FURTHER READING:

A First Course in Abstract Algebra, by John S. Fraleigh, Addison-Wesley, 1967.

Mathematical Reasoning, Anita Harnadek, Midwest Publications, 1972.

```
LIST
POLY4   Ø4:26 PM          16-DEC-75
10 PRINT"THIS PROGRAM CALCULATES THE ROTATIONS AND REFLECTIONS
11 PRINT"OF THE REGULAR POLYGONS .
12 PRINT "WHAT NUMBER OF SIDES (>2) DO YOU WANT?"
15 F$="!#  "
16 R$="R"
17 D$="D"
18 M$="M"
20 INPUT N
30 PRINT
40 IF 360/N=INT(360/N)THEN 80
50 PRINT "THERE IS NO REGULAR POLYGON OF";N;"SIDES WITH INTEGRAL ANGLES"
60 GO TO 20
70 DIM C(20,10)
80 D=0
85 REM CALCULATES THE ROTATIONS
90 FOR V=0 TO N-1
100 FOR I= 1 TO N
110 LET C(V+1,I)=I
120 NEXT I
130 FOR L= 1 TO N
140 Q=L+V
150 IF Q>N THEN Q=Q-N
160 C(V+1,L)=Q
170 NEXT L
180 PRINTUSING F$,R$,V;:PRINT"=";
190 FOR I= 1TO N
200 PRINT C(V+1,I);
210 NEXT I
220 PRINT
230 NEXT V
240 PRINT
245 REM CALCULATES THE FLIPS
250 IF N/2 = INT(N/2) THEN D=0 ELSE D=1
260 IF N02 = INT(N/2) THEN A=N/2 ELSE A=N
270 FOR I= 1 TO N
280 P= N+2-I
290 IF P>N THEN P=P-N
310 C(N+1,I)=P
320 NEXT I
330 V=1
335 K=N
340 GO SUB 450
350 FOR K=N+1 TO 2*N-1
360 FOR L= 1 TO N
370 C(K+1,L)= C(K,L)+2
380 IF K=3*N/2 THEN C(K+1,L)=C(K+1,L)+1
390 IF C(K+1,L)>N THEN C(K+1,L)=C(K+1,L)-N
400 NEXT L
410 V=V+1
420 GOSUB 450
430 NEXT K
440 GO TO 540
450 REM PRINTS OUT D'S AND M'S
460 IF V>A THEN D=D+1
470 IF V>A THEN V=1
480 IF D>1 THEN 540
490 IF D>0 THEN PRINTUSING F$,M$,V; :PRINT"=";:GO TO 510
500 PRINTUSING F$,D$,V;:PRINT"=";
510 FOR I=1 TO N: PRINT C(K+1,I);:NEXT I:PRINT
530 RETURN
540 PRINT
550 REM CALCULATES THE PRODUCT USING FIRST 2 NUMBERS
560 FOR K=1 TO 2*N
570 FOR K1= 1 TO 2*N
580 FOR I=1 TO 2
590 J=C(K,I)
600 T(I)=C(K1,J)
610 NEXT I
620 REM NEXT PART RECOGNIZES ANDPRINTS RESULT
630 IF T(1)<>N-1 THEN 650
640 IF T(2)=T(1)+1 THEN 720
650 IF T(2)=(T(1)+1)-INT(((T(1)+1)/N))*N THEN 720
660 IFT(1)/2=INT(T(1)/2) THEN T(1)= (T(1)+1+N)/2 ELSE T(1)=(T(1)+1)/2
670 IF N/2<>INT(N/2) THEN 700
680 IF T(1)<=N/2 THEN PRINTUSING F$,D$,T(1);:GO TO 730
690 PRINTUSING F$,M$,INT(T(1)-N/2);: GO TO 730
700 PRINTUSING F$,M$,T(1);
710 GO TO 730
720 PRINTUSING F$,R$,T(1)-1;
730 NEXT K1
740 PRINT:PRINT
750 NEXT K
760 GO TO 20
770 END
```

Introducing HEATHKIT® COMPUTERS

A new value standard in personal computing systems featuring two powerful computers with better software, full documentation and service support from the Heath Company.

Heath Company has been interested and involved with personal computing since we first marketed an analog computer system all the way back in 1957. This continuing interest, along with the recent technological advances that have brought personal computing to the forefront of the electronics marketplace, has given us the opportunity to think through the recent developments, and develop two "total design" computer systems that give the computer hobbyist, whether beginner or advanced, everything needed for REAL power, performance and reliability — at prices that give you MORE value and performance for your computer dollar!

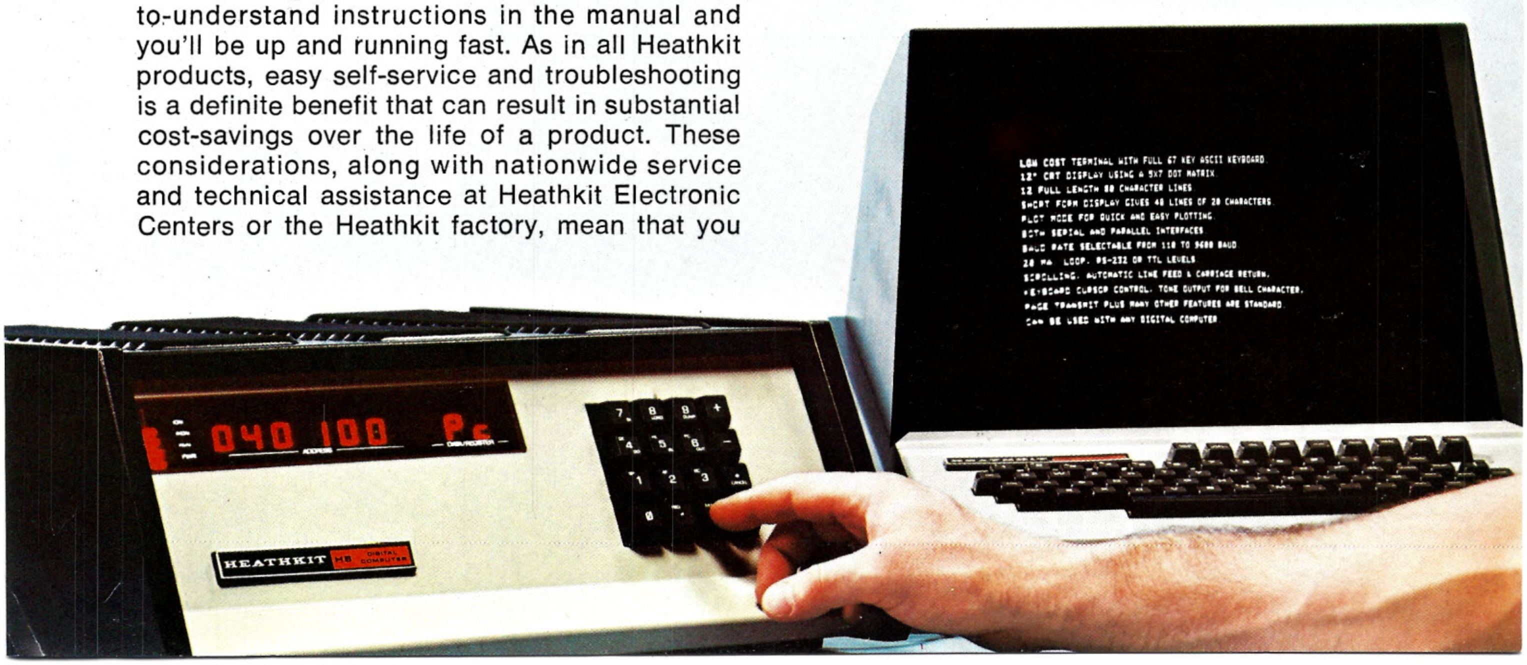
Total system design. The Heathkit computer line, both hardware and software, has been designed from the ground up to be a total computing system that meets all the needs of the computer hobbyist. The two mainframes are based on performance-proven well-documented MP modules, the 8080A and LSI-11. Using these CPU's was a conscious design decision, because of their proven performance, reliability and efficiency, and the tremendous amount of existing applications programs, documentation and source materials that are available. The Heath-designed CRT terminal, paper tape reader/punch, serial and parallel interfaces make total system setup easy and fast, and the Heath-designed software provided assures immediate usefulness and versatility.

Superior documentation. Heath Company is world-famous for the accuracy and clarity of its instruction manuals. The Heath computer line continues this well-deserved reputation. Assembly and operations manuals are written with easy-to-understand step-by-step instructions that leave nothing to chance. Simply follow the easy-to-understand instructions in the manual and you'll be up and running fast. As in all Heathkit products, easy self-service and troubleshooting is a definite benefit that can result in substantial cost-savings over the life of a product. These considerations, along with nationwide service and technical assistance at Heathkit Electronic Centers or the Heathkit factory, mean that you have the most reliable protection for your computer investment available anywhere.

System versatility. Both Heathkit computers offer full expansion potential to provide outstanding flexibility and adaptability to meet any application. Mass storage capability is available in both audio cassette and paper tape format on the H8 and in paper tape format on the H11 for added convenience. Additional memory expansion boards can be added to either unit, along with an expanding number of I/O devices.

Continuing Development. Heath will continue to design and develop new compatible products for their computer systems. Coming in the future will be — floppy disk storage, line printer, additional applications programs, and self-instructional courses in programming and assembly languages. All Heathkit computer users are eligible to join HUG (the Heath User's Group) and H11 customers are eligible to join DECUS, the Digital Equipment Computer User's Society.

We're confident you'll find the Heathkit computer line one of the most intelligent, sensibly developed and complete product lines available today. It offers you total versatility and expansion capability to go wherever your imagination and computing prowess take you. And, in the Heathkit tradition, it offers the best price/performance and reliability combination you'll find anywhere.



THE HEATHKIT H8 COMPUTER

A unique, value-packed computer featuring an "intelligent" front panel with built-in extended ROM monitor, octal entry keypad and digital readout, exclusive Heath bus, a pre-wired and tested 8080A-based CPU, and complete systems software at no extra cost!

$375.00

HEATH charg-a-kit REVOLVING CHARGE

HEATHKIT 8-BIT DIGITAL COMPUTER

A low-cost digital computer that's easier to build and to use! Features an intelligent front panel with keyboard entry and 9-digit display, a heavy-duty power supply with enough extra capacity for memory and I/O expansion and a 50-line fully buffered bus capable of addressing 65K bytes and a mother board with positions for up to 10 plug-in circuit boards. Includes BASIC, assembler, editor and debug software at no extra cost!

The Heathkit H8 computer is an 8-bit machine based on the popular 8080A chip. It is one of the lowest-cost general-purpose computers on the market, and thanks to Heath's exclusive design, one of the most versatile.

The interrupt controlled "intelligent" front panel gives you far more power and control than is found on conventional units with bit switches and indicators. The 16-digit keyboard allows octal data entry and control that's far faster and less error prone than binary switches. The 9-digit octal readout provides you with more information than conventional models too.

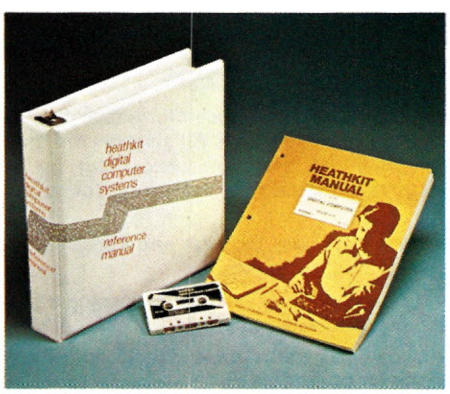
The octal keyboard and display emulate a true hardware front panel with complete access to memory, all registers and functions. The 9-digit seven-segment octal display has three readout modes: 6 digits of address and 3 digits data; 6 digits register data and 2 digits register identification; and three digits data with three digits port address. The front panel functions are defined by a panel monitor control program (PAM-8) stored in a 1K x 8 ROM on the CPU board. The complete access to 8080 internal circuits and functions makes the H8 an ideal trainer and learning tool.

Complete front panel functions include: display and alter of memory locations; display and alter of registers; dynamic monitoring of registers or memory during program execution; program execution control including break-point capability and single instruction step; automatic tape load and store through a built-in routine that allows programs to be loaded with a single button; and write or read any I/O port. The front panel of the H8 is so versatile it's like having a mini I/O terminal built right in!

Other features of the H8 front panel include status lights for power-on, run, monitor and interrupt enable; a built-in speaker for audible feedback on keyboard entry. The speaker also can be programmed for variable tones, permitting a variety of special effects to be generated.

The CPU board is fully wired and tested. It features the 8080A, clock, systems controller, ROM monitor and full bus buffering. Seven vectored interrupts are available on the bus for quick response to your I/O requests. A built-in clock lets you design and run in real time.

The H8 uses an exclusive, Heath-designed bus which incorporates many practical improvements over existing busses. The bus is fully buffered to reduce noise and crosstalk and is "glitch" free to eliminate timing problems. Three-state line drivers and receivers are used on all bus lines to eliminate loading problems. The 50 lines include address, data, control, clock and interrupt lines, plus all signals needed to support the 8080 MPU and virtually any I/O or memory accessory. The bus is implemented on a heavy-duty printed circuit mother board with wide, heavy copper foils for greater physical strength plus reduced crosstalk and noise. The board has 10 positions for installing connectors that accept the front panel, CPU, memory, I/O and accessory cards. All I/O bus connectors are included with the mother board for fast and easy expansion when you want it.

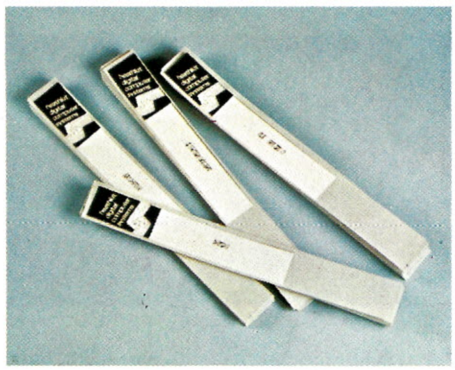


Comprehensive Heathkit assembly and operations manuals give you the superior documentation you NEED for a thorough understanding of your H8.

Systems software is supplied in audio cassette format.

The H8's built-in power supply is convection cooled for adequate ventilation without the use of noisy fans. Separate IC regulators provide distributed regulation with a heat sink on each circuit board for excellent heat dissipation. Power supplies of +8, −18 and +18 volts are provided to handle up to 32k memory plus three I/O interfaces. Switch-selectable 120 V, 60 Hz or 240 V, 50 Hz AC increases versatility.

The H8 includes all system software in 1200 baud audio cassette form at no extra charge. The Benton Harbor BASIC™ is an enhanced version of standard Dartmouth BASIC with unique statements and commands to extend usefulness. The efficient compression techniques of the Benton Harbor BASIC permit you to put more program in less space.



All H8 systems software is supplied in audio cassette form. Also available in paper tape (H8-15, page 5) at extra cost.

HASL-8 The Heathkit Assembly language is a 2-pass absolute assembler that lets you program with easily understood mnemonics and generates efficient machine language code. A minimum of 8K memory is required.

The TED-8 software is a line-oriented text editor used for generating source programs for the assembler or general word processing. Requires a minimum of 8K memory.

The BUG-8 a powerful terminal console debug program, is an enhanced and extended version of the front panel monitor program to allow entry and debugging of user machine language programs via an external terminal. Requires 3K memory plus user program.

The H8 is housed in a rugged, heavy-duty cabinet, 16¼" W x 6½" H x 17" D. Requires at least one H8-1 Memory.

Kit H8, Shpg. wt. 30 lbs. **375.00**

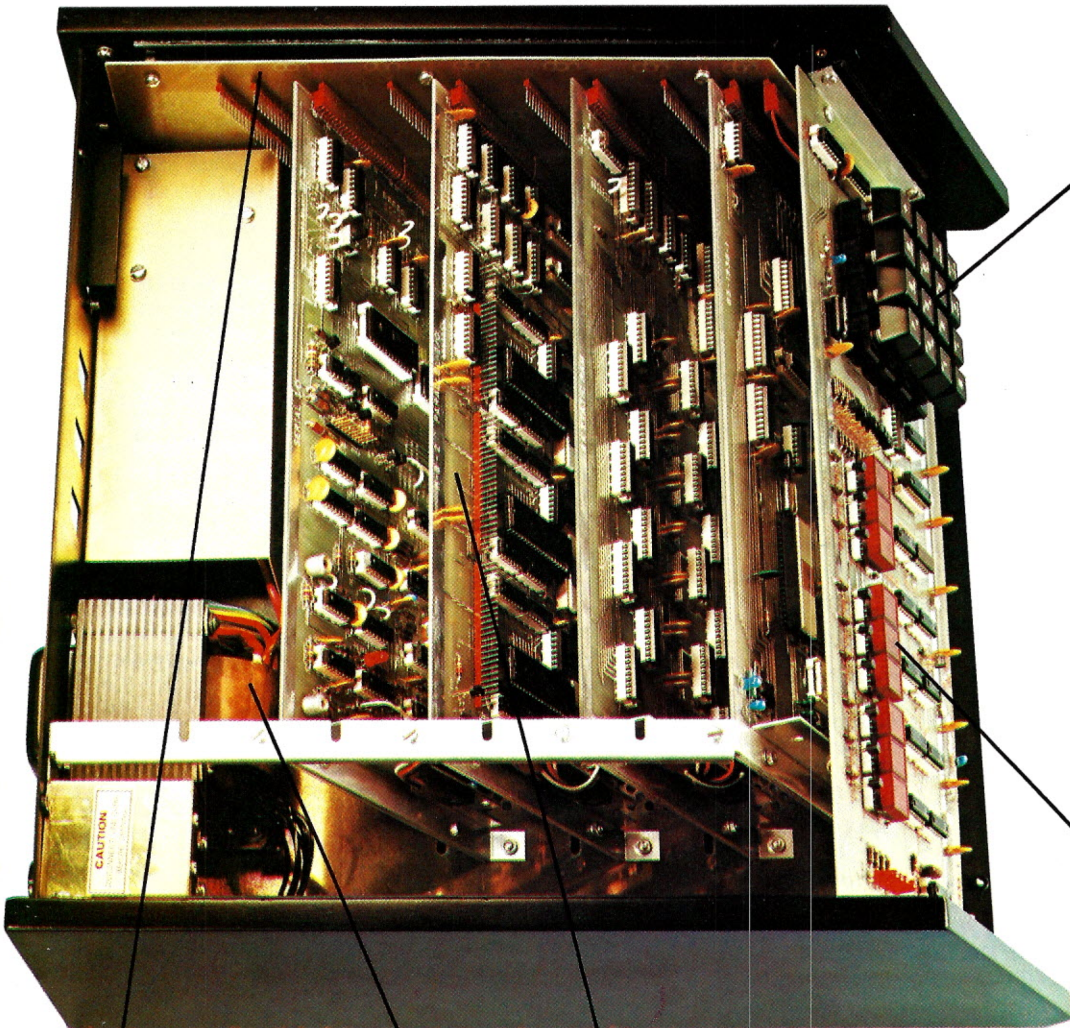
Suggested applications for the H8 computer: As a trainer—learn microprocessor operation, interfacing and programming. The powerful front panel lets you get at and use all parts of the unit.

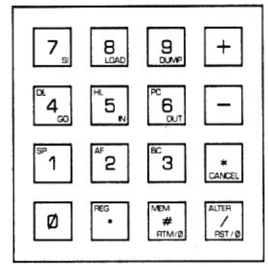
As an entertainment center—use game and other applications programs for entertainment the whole family can enjoy.

As a hobby computer—the H8 can be used to process any information you program into it—it's perfect for hobby experimentation and design. A variety of peripherals and interfaces let you use it with other equipment—run your Ham radio station, control your model railroad systems, etc.

As an educational system—the H8 is ideal for schools, community colleges, libraries, etc. Full H8 software permits teaching BASIC plus machine and assembly language programming.

As a home management center—use the H8 to keep telephone numbers, monitor your budget, keep your checkbook balanced, do your income taxes, inventory your personal belongings. There are hundreds of ways the H8 can make your life more convenient.





Its unique front panel keyboard makes the H8 the most powerful and sophisticated low-cost general-purpose computer available. Just take a look at these features!

- Direct-access to registers and memory even while program is running
- One button load and dump for fast, uncomplicated system startup
- Single instruction key lets you "step" through programs for easy debugging, program evaluation and learning
- Input/output keys let you communicate directly with any port

The unique Heath-designed 50-pin bus is implemented on a heavy-duty printed circuit board with heavy copper-foil bus lines. The 10-position mother board is complete with all connectors. The bus lines are fully buffered to eliminate noise and crosstalk, and "glitch-free" to prevent timing problems.

Modular circuit boards slide into the H8 mainframe for easy memory and I/O expansion, easy access for servicing. The boards are in a semi-vertical position with unconfined heat sinks to enhance convection cooling and improve heat dissipation.

Heavy-duty power supply, rugged steel chassis and securely mounted and braced circuit boards make the H8 a truly reliable and long-life machine.

Memory Display

High Order Address Location | Low Order Address Location | Data at Location 040 100

Register Display

High Order Contents | Low Order Contents | Register Identification

I/O Port Display

Data | Port Number

Unique Heathkit Software.

The Heathkit software supplied with the H8 computer has a number of features that make it easier to use and more practical than conventional systems. Automatic "command completion" simplifies typing; dynamic syntax checking instantly alerts you to errors and a special user configuration lets you really personalize your system. H8 software pushes the state-of-the-art a generation ahead — it's memory efficient to give you more computing power for your memory dollar, modular design for easy expansion, and thoroughly documented for easy programming and maximum effectiveness.

H8 "Intelligent" Front Panel

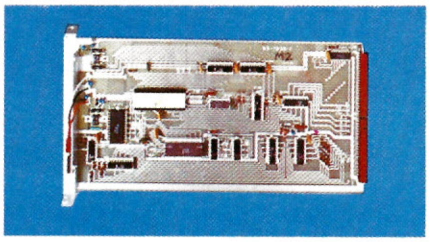
The H8 front panel digital readout is the most informative display available on any personal computer to date. All displays are continuously updated even while your program is executing, giving you instant access to registers and memory for direct monitoring of program activity.

MEMORY DISPLAY — Shows memory location and contents using 6 digits for address and 3 digits for data.

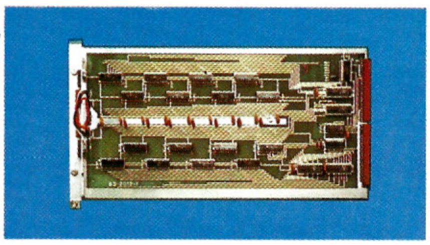
REGISTER DISPLAY — Shows CPU-register contents using 6 digits for data and 2 digits for register identification.

I/O PORT DISPLAY — Shows I/O port data and location using 3 digits for data and 3 digits for port address.

H8 ACCESSORIES, SOFTWARE AND MANUAL SET



The H8 CPU is fully wired and tested to insure quick and trouble-free system startup. It contains the performance proven 8080A microprocessor chip, a 1Kx8 ROM with monitor program for controlling the front panel and input-output (load-dump) routines. Other features of the CPU include: 7 vectored interrupts, DMA capability, crystal-controlled clock and fully buffered bus with three state drivers. Use of the 8080A, which has the largest software library of any microprocessor, along with Heath software and documentation, makes the H8 one of the most practical and immediately useful computers you can own.

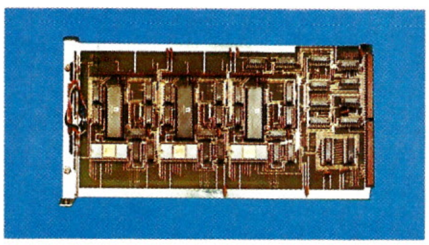


H8-1 Memory Board. 8Kx8 memory card supplied with 4K memory, plugs directly into H8 bus. Features maximum storage capacity of 8192 8-bit words. Uses modern 4Kx1 static memory IC chips for easy assembly and service. Access time, less than 450 nS. With on-board regulators, heat sinks and full buffering. Expandable to 8K memory with H8-3 chip set below.

Kit H8-1, Shpg. wt. 2 lbs. **140.00**

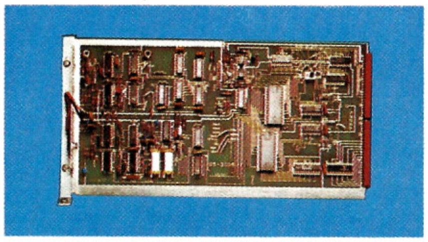
H8-3 Chip Set. Kit of eight 4K static memory IC's. Expands H8-1 to full 8K storage. With sockets.

Kit H8-3, Shpg. wt. 1 lb. **95.00**



H8-2 Parallel Interface. Connects H8 to any parallel device such as a paper tape reader/punch (required for H10) or line printer. Has three independent parallel ports, each with 8 bits input and 8 bits output and universal handshaking capability. Compatible with all Heath software. 390 μS maximum transfer time. With diode-clamped inputs, buffered outputs and full interrupt capability.

Kit H8-2, Shpg. wt. 3 lbs. **150.00**



H8-5 Serial I/O and Cassette Interface. Connects the H8 to serial devices such as the H9 video terminal (page 10) or the H36 DEC Writer II (page 12). Features jumper selectable data rate from 110 to 9600 baud, plus common input/output interfaces including 20 mA current loop and EIA RS-232C compatible levels. The cassette recorder interface permits the use of standard cassette recorders (Heathkit ECP-3801, page 12). Uses the popular Byte/Manchester or "Kansas City" standard recording format with a 300 or 1200 baud read/record rate. Control lines for remote start and stop of two cassette units allow separate record and playback for easy program or file editing. Also has full interrupt capability. LED test circuit for easy board setup and overall system servicing. Fully compatible with all Heath software.

Kit H8-5, Shpg. wt. 3 lbs. **110.00**

NOTE: Proper operation of the H8-5 is assured only if you use the Heath ECP-3801 cassette player/recorder and Heath-recommended recording tape (ECP-3802, page 12). Heath is not responsible for improper operation associated with other cassette units.

Extended Benton Harbor BASIC

Extended Benton Harbor BASIC is an enhanced and more powerful version of the BASIC supplied with the H8. It provides even faster operation and includes character strings, additional convenience commands and math functions, dynamic storage allocation, access to real time clock, keyboard interrupt processing, expanded error messages and recovery ability, LED display control and key pad support. A minimum of 12K memory is required to run this BASIC, 16K is preferred if full use is to be made of its capabilities.

H8-13 (1200 baud audio cassette)
Shpg. wt. 1 lb. **10.00**

H8-14 (fan fold paper tape)
Shpg. wt. 1 lb. **10.00**

Paper Tape Systems Software

A paper tape version of the systems software supplied with the H8 computer. It consists of four fan fold paper tapes, one each for Benton Harbor BASIC, HASL-8 assembler, TED-8 editor, and BUG-8 debug. For use with the H10 paper tape reader/punch or other paper tape I/O equipment.

H8-15, Shpg. wt. 1 lb. **20.00**

H8 Manual Set

Find out about the H8 before you buy! This manual set includes the complete assembly and operations manuals for the H8 Digital Computer, H8-1 memory card, H8-2 parallel interface, H8-3 4K memory expansion chip set, H8-5 serial and I/O cassette interface, H9 video terminal and H10 paper tape reader/punch. H8 software documentation covering monitor, editor, assembler, debug and BASIC is also included. In handsome 3-ring binder.

HM-800 Manual Set.
Shpg. wt. 11 lbs. **25.00**

The purchase price of the HM-800 manual set will be refunded when you buy the H8. Simply include HM-800 saleslip with your order.

Heath User's Group

You can get even more excitement and practical use from your H8 by joining HUG, the Heathkit User's Group. It will put you in contact with other Heathkit computer users, provide a program library and an informative newsletter to keep you up to date. A HUG application is enclosed with each Heathkit computer product. See page 12 for further details.

THE HEATHKIT H11 DIGITAL COMPUTER

Two of the finest names in modern electronics, Heath and Digital Equipment Corporation (DEC) combine to bring you the world's first 16-bit computer priced within reach of the general public!

$1295.00

The H11 and all its accessories will be available November 10th, 1977.

HEATHKIT®/DIGITAL EQUIPMENT CORPORATION®
H11 DIGITAL COMPUTER

Heath and DEC join forces to bring you mini-computer performance at a microcomputer price! The H11 features a fully wired and tested DEC KD11F board that contains the 16-bit LSI-11 CPU, 4096 x 16 read/write MOS semiconductor memory, DMA operation; and includes the powerful PDP-11/40 instruction set, PLUS Heath/DEC PDP-11 software. Equivalent commercial versions of the H11 would cost $1,000's of dollars more!

The new Heath/DEC H11 personal computer is one of the most powerful and sophisticated units available today! It combines the advanced, performance-proven hardware and software of the LSI-11 with Heath's expertise in kit design and documentation to bring you a personal computer of almost incredible power and flexibility. Equivalent commercial versions of the H11 would cost over twice as much, and you still wouldn't get the superior documentation and support of the H11!

The LSI-11 bus is a mechanically and electrically superior bus with 38 high-speed lines containing data, address, control and synchronization lines. Sixteen lines are used for time multiplexing of data and addresses. All data and control lines are bidirectional, asynchronous, open-collector lines capable of providing a maximum parallel data transfer rate of 833K words per second under direct memory access operation.

The 16-bit CPU functions are contained on four MOS LSI integrated circuit chips. These chips provide all instructions, decoding, bus control, and ALU functions of the processor. The CPU has eight general registers which serve as accumulators, index, autoincrement/autodecrement registers or stack pointer.

The KD11F memory is a 4096-by-16 MOS semiconductor memory composed of LSI 4K dynamic RAM chips. These chips require little power, provide fast access time, and are refreshed automatically by the processor's microcode. Additional memory cards can be added to expand memory capacity up to 20K in the H11 cabinet (32K words total).

The backplane/card guide assembly holds the microcomputer and up to six I/O and memory modules. All LSI-11 bus data, control, and power connections are routed on the printed circuit backplane to each module location. The backplane/card guides are fully compatible with all standard DEC LSI-11 accessories.

An efficient, well-designed switching power supply provides the required DC voltage for the LSI-11 as well as all accessory modules. The supply features overvoltage and overcurrent/short-circuit protection, power fail/automatic restart and a built-in fan for quiet cooling. The dual primary power configuration can be connected for 115 V, 60 Hz or 230 V, 50 Hz input power.

Has single-level, vectored, automatic priority interrupt, real-time clock input signal line, ODT/ASCII console routine/bootstrap resident in microcode for automatic entry into debugging mode, replacement of panel lights and switches with any terminal device generating standard ASCII code, and the ability to automatically commence operation through resident bootstrap routines.

The H11 is supplied with versatile PDP-11 software including editor, relocatable assembler, linker, absolute loader, debug program, I/O executive program, dump routines, BASIC and FOCAL (See details below). The software requires a minimum of 8K memory, with 12K to 16K total memory recommended for maximum capability. Rugged metal cabinet measures 6½" H x 19" W x 17" D. For 110/220 VAC, 50/60 Hz.

Kit H11, Shpg. wt. 34 lbs. **1295.00**
NOTE: See DEC software license form on page 15.

POWERFUL HEATH/DEC PDP-11 SOFTWARE AT NO EXTRA COST!

The H11 includes a sophisticated software system that lets you get your computer up and running with practical programming capabilities. This paper tape based software would cost over $1200 if purchased separately. A minimum of 8K memory is required to run the software. The programs include:

ED-11. Assists you in the creation and modification of ASCII source tapes, also used to write assembly language programs and for general text editing or word processing functions.

PAL-11S. Relocatable assembler converts ASCII source tapes into relocatable binary modules. This lets you create programs in small, modular segments for easier coding and debugging. These binary modules serve as inputs to LINK-11S.

LINK-11S. Link editor which links the modules created by the PAL-11S into a load module ready for execution on the H-11. The module is loaded into the H-11 via the Absolute Loader.

The H11 is complete with superior Heathkit documentation and versatile system software.

Absolute Loader. Loads absolute binary tapes into the H11 memory for execution.

ODT-11X. Lets you debug the programs which you have created. Permits modifying and controlling program execution "on the fly" for quick, efficient debugging.

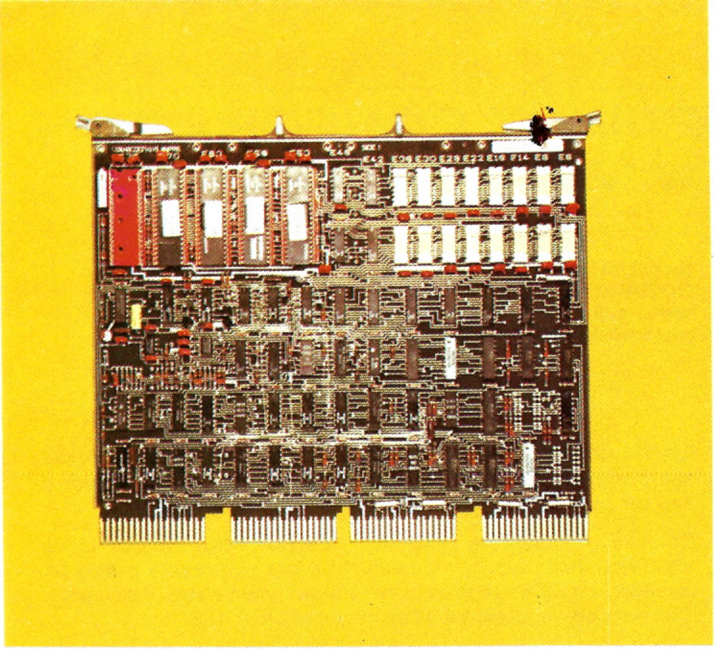
IOX. I/O executive program permits I/O programming without developing device-driving programs. Links to your programs using the LINK-11S. For use with high speed paper tape reader/punch and line printer.

DUMP-AB and DUMP-R. Lets you dump absolute binary contents of memory into the paper tape punch.

BASIC. DEC's powerful version of standard Dartmouth BASIC interpreter uses english-type statements and mathematical symbols to perform operations. Immediately translates, stores and executes the program. Includes string capability.

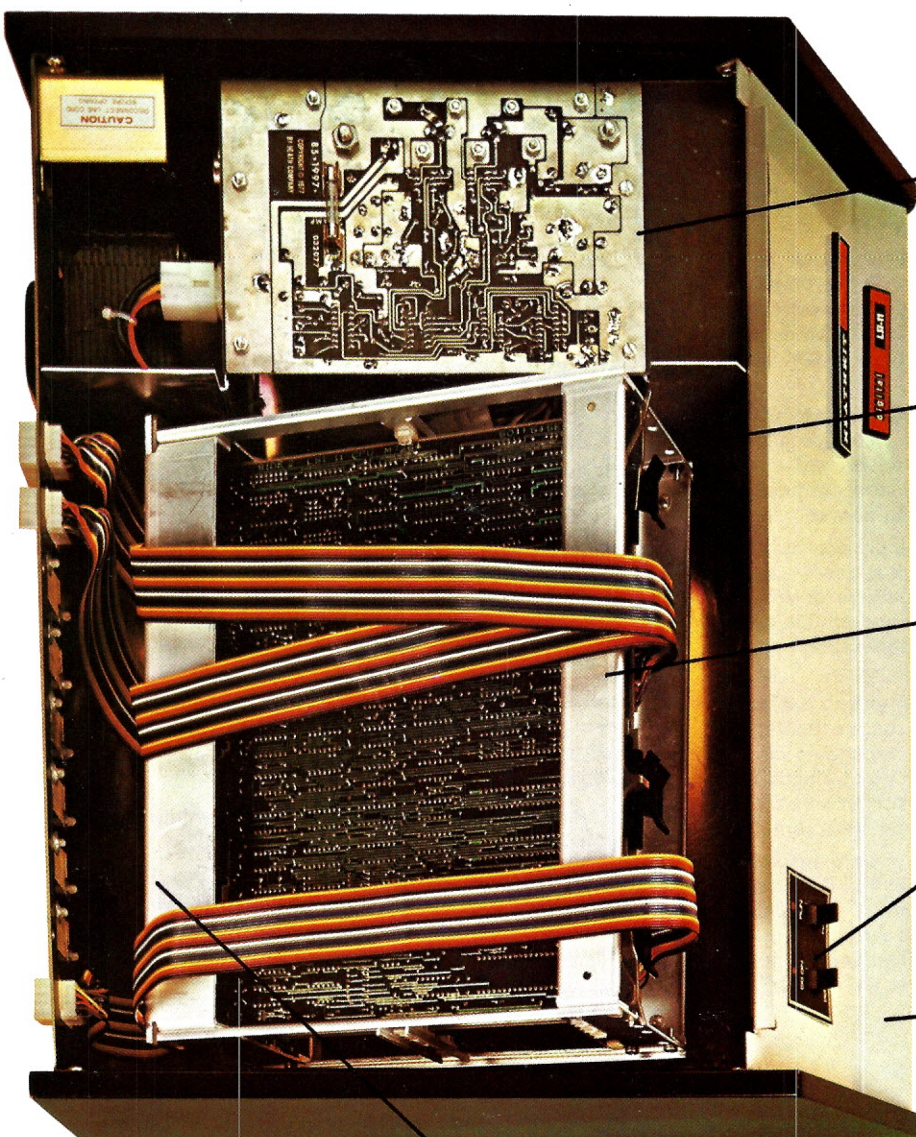
FOCAL™. DEC's own interpretive computer language which combines simplicity with computing power. Ideal for most scientific, engineering and math applications. FOCAL™ programs can be written and executed easily. Both 4K and 8K versions are included.

NOTE: H11 owners are eligible for membership in the Digital Equipment Computer User's Society (DECUS). This organization provides useful symposia, newsletters, program library and other useful information to help you get the most from your LSI-11 computer.



FULLY WIRED AND TESTED KD 11F BOARD

The "heart" of the H11 computer is the standard DEC LSI-11 microcomputer board. The 16-bit CPU functions are contained in four silicon gate N-channel MOS LSI integrated circuit chips for high reliability and superior performance. The 4096-by-16 read/write MOS semiconductor memory is composed of LSI 4K dynamic RAM chips that provide fast access time and require little operating power. The CPU executes the powerful PDP-11/40 instruction set with over 400 instructions. There are no separate memory I/O or accumulator instructions, so you can manipulate data in peripheral device registers as easily and flexibly as in memory registers. The LSI-11 board is supplied fully wired and tested to facilitate kit assembly and provide greater reliability and less chance of error.



Compact, efficient switching power supply uses less power to operate and generates less heat than conventional supplies. Overvoltage and overcurrent/short circuit protection, along with automatic power-up and power-down sequencing, provide high reliability and long life operation.

Built-in quiet-running fan provides efficient cooling and prevents heat buildup.

Card cage with backplane accommodates up to six accessory cards in addition to LSI-11. The card cage swings up for easy access and service even while the H11 is operating. Accessory boards slide directly into card guides with all connectors supplied.

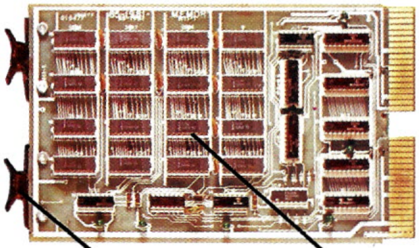
Front panel controls include DC power switch and run/halt switch. Status lights indicate processor activity.

Styled and sized to match Heathkit peripherals for total system continuity.

Rugged steel chassis and extra-thick backplane with heavy, solid connectors for added strength and years of superior performance.

The H11 and all its accessories will be available November 10th, 1977.

H11 ACCESSORIES, SOFTWARE AND MANUAL SET

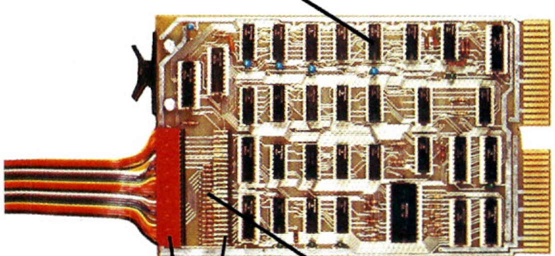


1

Card handles provide easy removal and insertion in card cage

Sixteen state-of-the-art 4K static memory chips for high density storage

All IC's are socketed for easy kit assembly, easy access for service or trouble-shooting

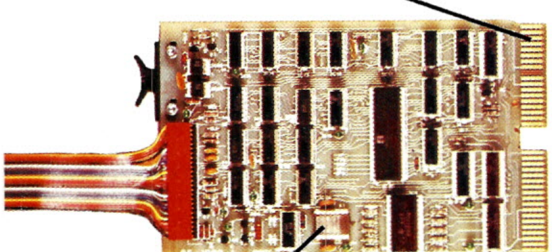


2

All inputs are diode-clamped for worry-free interfacing and system protection

Separate cables and rear panel connectors for high-byte and low-byte increase system flexibility

Gold-plated edge connectors maintain superior electrical contact for high reliability and long life



3

Quartz crystal and integrated baud rate generator has superior accuracy for reliable system interfacing

1 H11-1 4K Memory Expansion Module

Plugs into H11 backplane, adds 4K x 16-bit word capacity to H11 memory. Uses high-reliability 1Kx4 static MOS RAM chips. Access time is less than 500 nS. Has decode circuitry for operation on 4K address boundaries. Handle for easy removal and insertion. Compatible with PDP 11/03 and other LSI-11 backplane machines.

Kit H11-1, Shpg. wt. 2 lbs. **275.00**

2 H11-2 Parallel Interface

General-purpose parallel interface featuring 16 diode-clamped latched data input lines, 16 latched output lines, 16-bit word or 8-bit byte data transfers. Has LSI-11 bus interface and control logic for interrupt processing and vectored addressing; control status registers compatible with PDP-11 software routines. Four control lines for output data ready, output data accepted, input data ready and input data accepted logic operations. Maximum data transfer rate, 90K words per second under program control. Maximum drive capability, 25-ft. cable. Plugs into H11 backplane, can be used with DEC PDP-11/03 and other LSI-11 backplane machines. Also compatible with TTL or DTL logic devices. The H11-2 is required for interfacing the H11 to the H10 Paper Tape Reader/Punch.

Kit H11-2, Shpg. wt. 2 lbs. **95.00**

3 H11-5 Serial Interface

Universal asynchronous receiver/transmitter serial interface module for use between LSI-11 bus and serial devices such as the Heathkit H9 video terminal (page 10) or LA36 teleprinter (page 12). Has optically isolated 20 mA current loop and EIA interfaces; selectable baud rates of 50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800, 2400, 4800 and 9600. Plugs into H11 backplane, fully compatible with PDP 11/03 and other LSI-11 backplane machines. With all mating connectors.

Kit H11-5, Shpg. wt. 2 lbs. **95.00**

H11-6 Extended Arithmetic Chip

Adds powerful arithmetic instructions to the LSI-11, including fixed point multiply, divide and extended shifts plus full floating point add, subtract, multiply and divide. Helps minimize or eliminate arithmetic sub-routines, speeds up program execution and eases program development. Saves memory space too. 40-pin dual-inline package IC plugs into socket on KD11F board.

H11-6, Shpg. wt. 1 lb. **159.00**

Manual Set for H11 Computer

Includes complete assembly and operation manuals for the H11 Digital Computer, H11-1 4K memory board, H11-2 parallel interface, H11-5 serial interface, H9 CRT terminal, and H10 paper tape reader/punch. Also includes complete soft-

ware documentation — monitor, editor, assembler, linker, BASIC, FOCAL and related software. In handsome 3-ring binder.

HM-1100 Manual Set,
Shpg. wt. 12 lbs. **25.00**

NOTE: The price of the manual set can be deducted when you order an H11.

NOTE: DEC, DIGITAL, FOCAL and PDP are registered trademarks of Digital Equipment Corporation.

Special DEC Software License Requirement

H11 purchasers are required to fill out and sign the DEC license agreement on page 15. Please do so and include with your H11 order. Heath cannot ship merchandise without this license agreement.

THE HEATHKIT H9 VIDEO TERMINAL

One of the lowest-cost full ASCII terminals available anywhere—features a bright 12″ CRT display with twelve 80-character lines, 67-key keyboard, all standard serial interfaces, plus a fully wired and tested control board and a wiring harness for simplified assembly.

$530⁰⁰

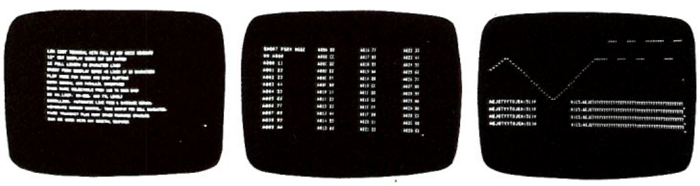
HEATH charg-a-kit REVOLVING CHARGE

H9 LONG AND SHORT-FORM VIDEO DISPLAY TERMINAL

The H9 video terminal is a general-purpose computer peripheral designed for use with the Heathkit H8 or H11 computers. It provides keyboard input and a CRT for the convenient entry and display of computer programs and data. The H9 can be used with any digital computer in dedicated stand-alone applications or in time-sharing systems.

Character format is standard upper case 5 x 7 dot matrix. The long form display is twelve 80-character lines. The short form display is forty-eight 20-character lines in four 12-line columns. The automatic line carryover feature executes line feed and return when line exceeds character count on both long and short form displays. A built-in oscillator/speaker generates a 4800 Hz tone and serves as audible end-of-line warning.

Auto-scrolling is featured in both long and short form. In the long form, as the line enters at bottom, the top line scrolls off-screen; in the short form, as new column enters from right, the left column scrolls off-screen. Auto-scrolling can be defeated with a front panel switch. The cursor mark indicates the next character to be typed for accurate positioning. Cursor control keys include up, down, left, right and home. Serial data baud rates are selectable from 110-9600. Baud rate clock output and reader control are available on the rear panel connector. The erase mode permits automatic full page erase or erase to end of line starting at cursor position. A transmit page function allows a full page to be formatted, edited and modified, then transmitted as a block of continuous data.



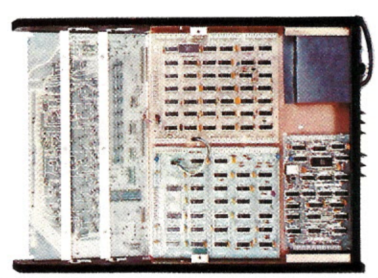
Long form—twelve 80-character lines

Short form—forty-eight 20-character lines

Plot mode—graphs, curves, simple figures

Three separate modes give the H9 real display versatility

The plot mode permits graphs, curves and simple figures to be displayed. Plotting can be accomplished via the keyboard or from external inputs.



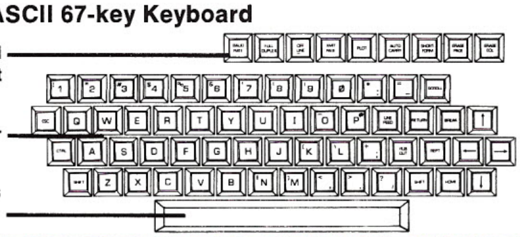
Control PC board is fully assembled and tested for added reliability and simplified kit assembly. A wiring harness with connectors helps reduce time-consuming point-to-point wiring.

The H9 serial interface provides EIA RS-232C levels, a 20 mA current loop or standard TTL levels. Parallel interfacing includes standard TTL levels, 8 bits input and 8 bits output and 4 handshaking lines.

Ultra-compact size, only 12½″ H x 15⅝″ W x 20¾″ D, makes the H9 ideal for desktop or console applications. For 110 VAC, 60 Hz or 230 VAC, 50 Hz.

Kit H9, Shpg. wt. 50 lbs. **530.00**

Full ASCII 67-key Keyboard

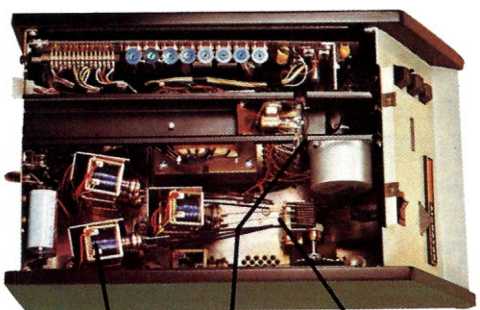


Function keys are positioned away from characters to prevent miskeying and error.

Standard typewriter keyboard for easy, more accurate input.

Wide, easy-to-use space bar aids accurate typing.

H10 DELUXE PAPER TAPE READER/PUNCH



Heavy-duty solenoids for reliable, long-life operation

Precision machined punch block for accurate, consistent punched holes

Advanced reader design with stepper motor and solid-state sensors for accurate reading

The H10 is a complete paper tape reader/ punch mass storage peripheral using reliable low-cost paper tape. It's fully compatible and styled to match with the H8 and H11 computers. It also works reliably with any other computer through a parallel interface. The H10 uses standard 1" wide roll or fan-fold 8-level paper tape. Standard punched paper tape gives you the reliability, durability and trouble-free handling you need for effective mass storage of programs and data.

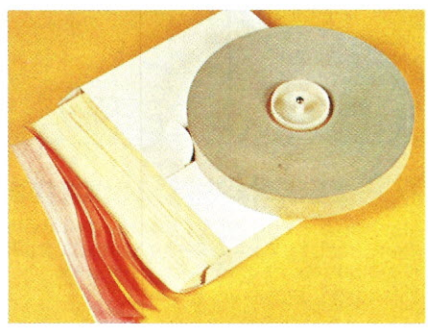
The reader reads tape at a maximum rate of 50 characters per second. A full sensitivity adjustment on each channel permits any color, thickness, quality (oiled or unoiled) paper tape to be used. Sensitive photo Darlington transistors and an incandescent lamp reader head provide reliable reading. The powerful stepper motor drive insures accurate tape positioning and movement.

The punch operates at a maximum speed of 10 characters per second. Precise ratchet/solenoid drive and reliable solenoid control of punches provide high-accuracy punching. The precision die-block punch head gives you positive and consistent punching.

Controls include power on-off, read and punch start. A feed control feeds blank paper tape through the punch to produce leader tape. A copy control on the rear panel permits tape being read to be duplicated by the punch for efficient and accurate tape copying.

Interface has parallel 8-bit input bus for punch, parallel 8-bit output bus for reader, standard TTL logic levels and handshaking lines for both reader and punch. A rear panel 24-pin interface connector and mating cable are supplied. The H10 is fully compatible with Heathkit H8 and H11 computers when the appropriate parallel interface accessories are used. It can also be interfaced with other computers with parallel interface facility.

Accessories include holder for roll paper tape, chad collector tray, and collector box for fan-fold tape. With 8" roll (900 ft.) blank paper tape.



Styled to match the Heathkit H8 and H11 computers. Cabinet with metal top and rugged steel chassis, 12⅝" H x 9¾" W x 19⅝" D. For 110-130 VAC, 60 Hz, or 220-240 VAC, 50 Hz.

Kit H10, Shpg. wt. 29 lbs. **350.00**

H10-2, Three Blank Rolls Paper Tape, each 8" diameter, 900 ft. min.

H10-2, Shpg. wt. 5 lbs. **10.00**

H 10-3, Three Boxes Blank Fan-fold Tape. Approx. 1000 ft. each.

H10-3, Shpg. wt. 5 lbs. **10.00**

ECP-3801 Cassette Recorder Storage Device

Has volume and tone controls, pushbuttons for record, play, rewind, fast forward, stop and eject, built-in 3-digit counter with reset button. Factory wired, not a kit.

ECP-3801, Shpg. wt. 6 lbs. **55.00***

Heath recommended high output, low noise, premium grade audio recording tape. Pack of three 30-minute blank cassettes.

ECP-3802, Shpg. wt. 1 lb. .... **per pack 5.00**

*NOTE: Proper operation of the H8-5 and H8 software is assured only when the ECP-3801 cassette recorder and ECP-3802 tape is used. Heath does not assume responsibility for improper operation resulting from the use of any other cassette units.

HUG®–the Heathkit User's Group

Our new user's group brings you in contact with other Heathkit computer owners and users, provides a newsletter, a program library, new product information and hardware/software ideas. Membership in HUG is a useful, practical way to get the maximum enjoyment and benefit from your Heathkit computer system. Here's what you get:

- 1 year subscription to the quarterly newsletter
- Software library allowing you to submit programs and obtain programs submitted by others. A modest fee will be charged for software duplication.
- An attractive 3-ring binder to hold newsletters, software documentation and other materials.
- Program submission forms • Software library catalog
- HUG membership list • Credit toward purchase of software

Dues are $14.00 for one year. Complete details of HUG membership are included with every Heathkit computer product. H11 owners are also eligible for membership in DECUS, see page 7 for details.

LA36 DEC Writer II Keyboard Printer Terminal

The famous LA36 DEC Writer II with true 30-cps throughput, variable-width forms handling, 128-character upper/lower case set, and extra-quiet operation. Fully assembled, factory tested and ready to use!

The LA36 is an advanced technology teleprinter offering fast, reliable operation at one of the best price/performance ratios in the industry. It features a 7x7 dot matrix print head for crisp, clear character formation; switch-selectable 10, 15 and 30 cps printing speeds; variable width forms handling from 3 to 14⅞" wide; adjustable right and left hand tractors for precise margin positioning; half or full duplex operation; ANSI-standard multikey rollover and a typewriter-like keyboard.

The precision-designed stepper motor paper feed has fine vertical adjustment for accurate forms placement. LA36 will handle up to 6-part forms with a .020" maximum pack thickness. Print format is 132-column, with 10 characters per inch horizontal spacing and 6 lines per inch vertical spacing. Uses the entire 128 character ASCII upper/lower case set with 95 printable characters. A CAPS-lock key simplifies data entry. A parity check on output prints a replacement character, strappable to odd, even, or none with mark or space. A last-character visibility feature moves the head four columns to the right when printing stops, returns to proper position when printing is resumed.

The integral 20 mA current loop interface makes the LA36 compatible with both the H8 and H11 computers, as well as all other hobby and personal computers. Operates on 90-132 VAC or 180-264 VAC for reliable performance even under brown-out conditions. With connecting cable and integral stand for easy setup. Overall size, 27½" W x 33¼" H x 24" D.

H36 (LA36 DEC Writer II) Shipped Motor Freight, prepaid to your nearest terminal within the Continental U.S. Include your phone number on order for notification of arrival. Arrangements for home delivery at extra charge at your option. NO C.O.D ORDERS ACCEPTED. **$1495.00**

H36-1 Fan-fold paper for H36. Standard 14⅞" x 11" white and green, single part, lined paper. 3450 sheets per carton.

H36-1, Shpg. wt. 50 lbs. **30.00**

H36-2 EIA Interface. Provides EIA RS232-C or CCITT-V24 interface for LA36. Includes auto answer, timed disconnect and half/full duplex logic. Straps are available to send timed break (230 mS), 3000 mS long space, forced disconnect or do nothing (stops printing, discards data) on a paper-out condition. Modem controls and a 9-ft. cable with 25-pin data-set type connector are also supplied. Factory wired, not a kit.

H36-2. Shpg. wt. 1 lb. **65.00**

APPLICATIONS SOFTWARE–COMING SOON!

Both the H8 and H11 Digital Computers are supplied with complete systems software that provide you with everything you need to develop your own specific applications programs. However, you can make your computer immediately useful by using the programs below. These programs represent the beginning of a complete series of application software packages that will allow you to get immediate value from your computer system without a time consuming software development effort on your part. Described below are a series of game packages that make your computer an excellent source of entertainment and leisure time activities.

BLACKJACK. An interactive program game that allows four players to play the card game blackjack on the computer. The computer performs all of the functions of the dealer and keeps track of player progress, winnings and losses. The program is written in and runs under extended BASIC and requires a minimum of 16K of RAM in the H8 and 8K in the H11. Standard Las Vegas casino blackjack rules apply.

BIORHYTHM. This popular applications program computes standard biorhythm information and plots sinusoidal curves of your physical, emotional, and intellectual characteristics over a given time period. The biorhythm program will show you your ups and downs and will tell you your good and bad days. It will help you plan your activities. While this program is not a game, it is an entertaining activity that you and your friends and family will enjoy. The program runs under extended BASIC and requires 16K of RAM in the H8 and 12K RAM in the H11.

STARTREK. Startrek is perhaps the most popular computer game available. It allows you to guide, control and command the Starship Enterprise in its travels through the galaxy, fighting Klingons and solving a variety of problems. A truly challenging, sophisticated and entertaining computer game. Runs on the H8 or H11 computers with 8K of RAM or more.

GAME SET #1. This software package lets you play 8 popular computer games. These games include Craps, Orbit, Tic Tac Toe, Nim, Hexapawn, Hangman, Hmrabi, and Derby. 8K RAM or more is required on either the H8 or H11. These games will provide hours of entertainment for you and your family.

GAME SET #2. Another popular game package for the H8 and H11 computers. Contains 8 popular computer games including bagles, slot machine, gomoko, yahtze, apollo, gunner, wumpus, and cube.

AVAILABILITY. Blackjack, Biorhythm and Startrek will be available after October, 1977. Game Set #1 will be available November, 1977 and Game Set #2 available, February, 1978.

Order a complete Heathkit computer system and SAVE!

The ECP-3801 is the Heath-recommended cassette recorder/player for use with the H8 computer software. See opposite page for complete description.

Choose any of the Heath-recommended systems shown here, the specially-priced HS-11 system below, or "roll your own" with a selection of products you choose. Any way you do it, you'll get a top-value, high-performance system, and you'll SAVE 5%!

Here's how to qualify for the 5% computer systems discount:

1. Select either the H8 or H11 and one major peripheral (H9, H10 or LA36).
2. Choose the I/O interface, memory and software accessories you need.
3. Specify each in the spaces provided on the order blank.
4. Deduct 5% from the total price of the products (excluding shipping and handling charges).

*Systems illustrated already have discount prices calculated for you.

From time to time, Heath will offer specially priced total systems such as the HS-11 below. These systems will be discounted even deeper than 5% to provide you with even MORE value for your computer dollar!

SYSTEM ONE

The minimum recommended H8 system

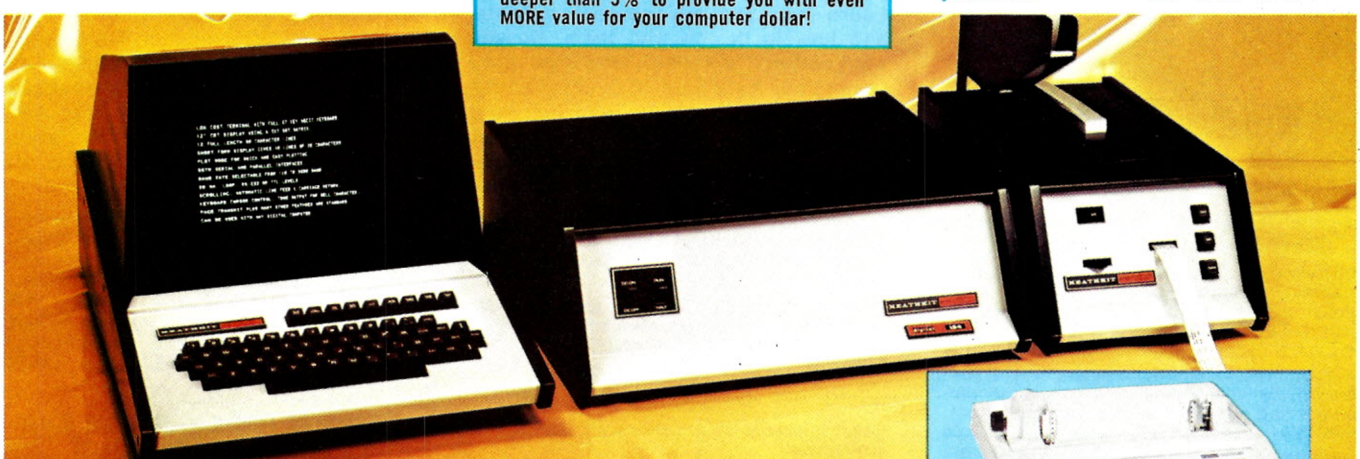
H8 Computer	$375
H8-1 4K Memory	140
H8-3 4K Chip Set	95
H8-5 Serial I/O and Cassette Interface	110
H9 Video Terminal	530
ECP-3801 Cassette Recorder/Player	55
If purchased separately,	$1305.00

Heath System price is **$1239.75***

SYSTEM TWO

H8 Computer	$375
Two H8-1 4K Memories	280
Two H8-3 4K Chip Sets	190
H8-5 Serial I/O and Cassette Interface	110
H8-13 Extended BASIC in Cassette Format	10
H9 Video Terminal	530
ECP-3801 Cassette Recorder/Player	55
If purchased separately,	$1550.00

Heath System price is **$1472.50***



SYSTEM THREE

The minimum recommended H11 system

H11 Computer	$1295
H11-1 4K Memory	275
H11-2 Parallel Interface	95
H11-5 Serial Interface	95
H9 Video Terminal	530
H10 Paper Tape Reader/Punch	350
If purchased separately,	$2640.00

Heath System price is **$2508.00***

SYSTEM FOUR

SAVE EVEN MORE on this specially-priced Heath HS-11 complete computer system!

H11 Computer	$1295
H11-1 4K Memory	275
H11-2 Parallel Interface	95
H11-5 Serial Interface	95
H10 Paper Tape Reader/Punch	350
H36 LA36 DEC Writer II	1495
If purchased separately,	$3605.00

$3350.00*

YOU SAVE $255!

NEW HEATHKIT SELF-INSTRUCTIONAL COMPUTER COURSES

These Heathkit self-instructional courses are designed to help you get the most from your computer investment, whether you buy your computer hardware from Heath or anywhere else. While many predeveloped software programs are available, the only way to realize the full value of your personal computer is to learn programming yourself. These courses use the PROVEN Heathkit individual learning techniques to give you a thorough understanding of programming, even if you have no prior knowledge or experience. They'll show you exactly how to make your personal computer system really personal!

BASIC PROGRAMMING SELF-INSTRUCTIONAL COURSE

This course teaches you how to program your computer using the popular BASIC language. BASIC (Beginner's All-Purpose Symbolic Instruction Code) is essential for hobby and personal computing; it is also widely used in education and business. The course covers all formats, commands, statements and procedures plus the creative aspects of computer programming, so you can make practical use of it in solving problems and creating your own unique programs. Like other Heathkit self-instructional courses, it uses programmed instructions backed by practical hands-on computer experiments and demonstrations to reinforce and personalize the text material. An optional final exam (passing grade 70%) brings you a Certificate of Achievement and 3.0 Continuing Education Units*. While the BASIC course is keyed to Heathkit computers, it is also equally applicable to any computer system using BASIC. Available after Oct. 20th, 1977.

EC-1100, Shpg. wt. 6 lbs. **29.95**

*Continuing Education Units (CEU's) are nationally recognized means of acknowledging participation in non-credit adult education.

COMING SOON!

8080 Programming: Teaches you the machine and assembly language used with 8080-based computers. Shows you how to use the editor, assembler and debug software to create efficient programs. Ideal for the H8 and other 8080A based machines. Coming soon.

LSI-11 Programming: Shows you how to use editor, assembler, linker, debug and executive I/O software to create efficient programs. Applicable to H11 and most other Digital Equipment Corporation PDP-11 series computers. Coming soon.

COMING IN OCTOBER! MICROPROCESSOR SELF-INSTRUCTIONAL COURSE

Learn how microprocessors operate and how to design with them. Covers applications, machine language programming, hardware I/O interfacing and much more. The course includes all IC's needed to perform exciting experiments. The microprocessor trainer used with the course features the popular 6800 microprocessor plus 256 bytes of RAM, a 1K ROM monitor, 6-digit hexadecimal display and hexadecimal keyboard. The Heathkit microprocessor course/trainer combo is the fast, easy low-cost way to learn about these important devices. Watch for it in our next catalog!

GENERAL COMPUTER BOOKS

Microcomputer Dictionary and Guide (Matrix). Comprehensive source of definitions and basic information on computers and related topics. A super reference source. A must for your library. **EDP-218** **17.95**

Introduction to Microcomputers Vol. I (Osborne). Excellent introduction to microcomputers and fundamental computer concepts. **EDP-224** **7.50**

Introduction to Microcomputers Vol. II (Osborne). Complete descriptions of all popular microprocessors, 8080, 6800, 6502, SC/MP, Z80, F8, 2650, etc. Good reference. **EDP-225** **12.50**

How to Buy and Use Minicomputers and Microcomputers (Sams). A fundamental text on mini/micro operation and application. **EDP-227** **9.95**

TV Typewriter Cookbook (Sams). Good text explaining I/O terminals, interfacing, etc. **EDP-226** **9.95**

8080 BOOKS

Build a library to support your H8 computer or any 8080 based machine.

Bugbook III (E & L). Superior reference source on 8080 interfacing and programming. Includes experiments. **EDP-231** **15.00**

Bugbook IIA (E & L). Serial I/O concepts and terminal interfacing. **EDP-232** **5.00**

8080 Programming for Logic Design (Osborne). Programming and Using the 8080 microprocessor. **EDP-229** **7.50**

8080 Software Gourmet Guide and Cookbook (Scelbi). Excellent source for 8080 programs and subroutines. **EDP-228** **9.95**

Practical Microcomputer Programming: 8080 (Northern Technology). Machine/Assembly programming concepts for the 8080. **EDP-235** **21.95**

6800 BOOKS

Great reference sources for your 6800 based computers.

6800 Programming for Logic Design (Osborne). Programming and using the 6800 microprocessor. **EDP-230** **7.50**

6800 Software Gourmet Guide and Cookbook (Scelbi). Excellent source of commonly used 6800 programs. **EDP-233** **9.95**

6800 Microprocessor Applications Manual (Motorola). Comprehensive review of typical 6800 applications, design solutions, etc. **EDP-244** **25.00**

6800 Microprocessor Programming Manual (Motorola). Programming principles and examples for the 6800. **EDP-245** **10.00**

H11/LSI-11/PDP-11 BOOKS

Here are several important reference sources to help you get the most value from your H11 Computer.

Minicomputer Systems: Organization and Programming (Prentice-Hall). Good basic text. Emphasis on the PDP-11. **EDP-238** **17.95**

PDP-11 Programming (Algonquin). A programmed instruction text teaching the concepts of PDP-11 operation and programming. **EDP-239** **5.00**

The Minicomputer in the Laboratory (Wiley). Operation, programming and applications of PDP-11 computers. **EDP-246** **19.50**

GENERAL PROGRAMMING AND APPLICATIONS BOOKS

Assembly Level Programming (Lexington). Good basic book on assembly language programming of small computers. **EDP-236** **14.95**

101 BASIC Computer Games (DEC). A classic. Have fun with your computer. **EDP-237** .. **7.50**

BASIC Software Library, Vol. I. Complete lists of BASIC applications programs bookkeeping, games, pictures (graphics). **EDP-240** **24.95**

BASIC Software Library, Vol. II. Math, engineering, plotting and statistical programs in BASIC. **EDP-241** **24.95**

BASIC Software Library, Vol. III. Advanced business applications programs in BASIC. **EDP-242** **39.95**

BASIC Software Library, Vol. IV. Games and business applications programs in BASIC. **EDP-243** **9.95**

BASIC Software Library, Vol. V. Games, graphics, and useful math programs in BASIC. **EDP-251** **9.95**

To remove order form cut out entire page along dotted line.

HEATH Schlumberger

Heath Company, Dept. 355-330
Benton Harbor, Michigan 49022

ORDER FORM
Agreement

Gentlemen: Please send me the Heathkit Computer Products I have checked below. I understand that if I order products designated for future delivery Heath will do their best to ship within 30 days of those availability dates.

- ☐ H8 Computer at $375.00 each plus $5.40 shipping and handling.
- ☐ Qty. ________ H8-1 4K Memory(s) at $140.00 each plus $1.15 shipping and handling.
- ☐ Qty. ________ H8-3 4K Chip Set(s) at $95.00 each plus $1.15 shipping and handling.
- ☐ H8-2 Parallel Interface at $150.00 each plus $1.30 shipping and handling.
- ☐ H8-5 Serial I/O Cassette Interface at $110.00 plus $1.15 shipping and handling.
- ☐ H8-13 Extended BASIC Cassette at $10.00 each plus $1.15 shipping and handling.
- ☐ H8-14 Extended BASIC on paper tape at $10.00 each plus $1.15 shipping and handling.
- ☐ H8-15 Paper Tape Systems Software for H8 at $20.00 each plus $1.15 shipping and handling.
- ☐ HM-800 Manual Set at $25.00 each plus $2.37 shipping and handling.
- ☐ H11 Computer at $1295.00 each plus $5.52 shipping and handling.
- ☐ Qty. ________ H11-1 4K Memory(s) at $275.00 each plus $1.15 shipping and handling.
- ☐ H11-2 Parallel Interface at $95.00 each plus $1.15 shipping and handling.
- ☐ H11-5 Serial Interface at $95.00 each plus $1.15 shipping and handling.
- ☐ Please send the following Computer Books (order must total $10.00 minimum) postpaid (Print numbers plainly.)

____

____

- ☐ H11-6 Extended Arithmetic Chip at $159 each plus $1.15 shipping and handling.
- ☐ HM-1100 Manual Set at $25.00 each plus $2.50 shipping and handling.
- ☐ H9 Video Terminal at $530 each plus $7.64 shipping and handling.
- ☐ H10 Paper Tape Reader/Punch at $350 each plus $4.96 shipping and handling.
- ☐ H10-2 Three Rolls Paper Tape at $10.00 plus $1.15 shipping and handling.
- ☐ H10-3 Three boxes Fan-fold Paper Tape at $10.00 plus $1.15 shipping and handling.
- ☐ H36 DEC Writer II at $1495 (No COD's, see page 12 for shipping information.)
- ☐ H36-1 Fan-fold Paper at $30.00 plus $7.76 shipping and handling.
- ☐ H36-2 E1A Interface at $65.00 each plus $1.15 shipping and handling.
- ☐ ECP-3801 Cassette Recorder/Player at $55.00 each plus $1.69 shipping and handling.
- ☐ ECP-3802 Cassette Recording Tape. Pkg. of three at $5.00 plus $1.15 shipping and handling per pkg.
- ☐ EC-1100 BASIC Programming Course at $29.95 plus $1.69 shipping and handling. Available after October.
- ☐ HS-11 Special Priced Complete System at $3350 each plus $14.00 shipping and handling.

Note: The H11 and all its accessories will be available November 10th, 1977.

HEATH REVOLVING CHARGE PLAN

charg-a-kit

You may purchase Heathkit products on our convenient Revolving Charge Plan. No money down and up to two years to pay. Up to $1500 maximum account balance.

☐ I would like to open a Heath Revolving Charge Account. Please rush me the necessary application forms when you receive this order.

I enclose my ☐ check ☐ money order for $____________ (Michigan residents add 4% sales tax.)

Or charge to my ☐ Visa/Bankamericard ☐ Master Charge

Acc't No.____________________ Exp. Date____________

If Master Charge, include Code No.____________________

☐ Add on to or reopen my existing Heathkit Charge Account. No.____________
($1500 maximum account balance.)

Signature____________________________________
(necessary to ship merchandise)

Name____________________________________
Please print plainly

Address____________________________________

City____________________ State____________ Zip____________

BR-119 All prices net F.O.B. Benton Harbor, Mi. Prices and specifications subject to change without notice.

HEATH/DEC Software License Agreement:

This form MUST accompany your H11 computer order.

CUSTOMER SUBLICENSE GRANT

HEATH COMPANY (hereinafter referred to as HEATH) pursuant to a license agreement with Digital Equipment Corporation (hereinafter referred to as DIGITAL) does hereby grant to CUSTOMER a non-transferable and non-exclusive sublicense to use the Binary Software Program(s) PTSP-11 Paper Tape System, FOCAL/PTS Language Processor, BASIC/PTS Language Processor (hereinafter singularly and/or collectively referred to as "Software") on the following terms and conditions.

DESCRIPTION OF SOFTWARE:

Software is furnished to CUSTOMER for use on a single CPU only and may be modified, or copied (with the inclusion of DIGITAL'S copyright notice) only for use on such CPU. The CUSTOMER shall not provide or otherwise make available the Software or any portion thereof in any form to any third party without the prior approval of DIGITAL. Title to the ownership of the Software shall at all times remain with DIGITAL.

DIGITAL AND HEATH JOINTLY AND SEVERALLY DISCLAIM ALL IMPLIED WARRANTIES WITH REGARD TO THE SOFTWARE LICENSED HEREUNDER, INCLUDING ALL WARRANTIES OF MERCHANTABILITY AND FITNESS; and any stated express warranties are in lieu of all obligations or liability on the part of either DIGITAL or HEATH for damages, including but not limited to special, indirect or consequential damages arising out of or in connection with the use or performance of the Software licensed hereunder.

This Sublicense Grant, the licenses granted hereunder and the Software may not be assigned by the CUSTOMER without prior written consent from DIGITAL. No right to reprint or copy the Software, in whole or in part, is granted hereby except as otherwise provided herein.

HEATH COMPANY

By William E. Johnson

CUSTOMER

By____________________________
Customer's Signature

Heathkit is the name to look for in quality electronic kits

Heath Company, located on the shores of Lake Michigan near Benton Harbor, is the world's largest manufacturer of electronic kits. Heathkit products are marketed primarily through the Heathkit Mail Order Catalog, published 5 times a year, and a nationwide chain of Heathkit Electronic Centers (Units of Schlumberger Products Corporation).

In addition to this new line of personal computing products, Heath Company offers nearly 400 other electronic kits of virtually every description. Product categories include Amateur Radio, Automotive Accessories, Stereo Hi-fi Components, Test Instruments, Color TV, Boating and Marine Accessories, Leisure and Home Improvement Products, and our award-winning self-instruction programs. Every Heathkit product is supplied with a comprehensive step-by-step instruction manual that tells you all you need to know, from unpacking the kit to plugging it in. These manuals are world-famous for their clarity, accuracy and precision. Let Heath show you how easily you can put together superior quality electronic products. Send for your FREE copy of our latest catalog!

Heathkit Computer Systems are also sold and serviced through 47 Heathkit Electronic Centers nationally (Units of Schlumberger Products Corporation) at slightly higher prices. Check the white pages of your telephone directory for the Heathkit Electronic Center nearest you.

Canadian Customers: Contact the Heathkit Electronic Center in most major cities or write for Canadian Prices to:

Heath Company, A Division of Schlumberger Canada, Ltd.
1480 Dundas Street E.
Mississauga, Ontario L4X 2R7

Printed in U.S.A.

A COMPARISON OF SOFTWARE SYSTEMS

In Volumes 1 and 2 of *Creative Computing* we ran an in-depth comparative review of every book on BASIC currently in print. While many sources review books, few, if any, compare them to each other. This group review was very well received.

Going one step beyond book reviews we've now embarked on a project to publish in-depth, comparative reviews of software and hardware/software systems. As with the books, these won't be based on manufacturer's literature or published specifications, but rather on our own exhaustive tests.

The first review of several BASIC language interpreters from this project appears in this issue. However, we are still in the formulation stage with the overall effort.

BASIC Language Comparisons

On what basis should a BASIC interpreter be compared? Number of statements? String functions? Matrix manipulation? Graphics commands? Level of FOR loop nesting? Richness of statements? File handling? Formatting? Timing functions? Probably all of the above and more. But with what emphasis? Frankly, that all depends upon your expected end use.

We've tried to come up with a number of reasonable comparative variables, although as this series of tests progresses, we'll surely think of more. If you have some ideas, please let Steve North or me know.

Hardware/Software Systems

Some time ago I got it into my head that it would be neat to devise a test routine that could be used on everything from the smallest microcomputer (or calculator for that matter) to the largest, giant number-cruncher. With that in mind I recalled "Different Digits," a problem we published back in Vol. 1, No. 2. Geoffrey Chase, D.O.M. at the Abbey, Portsmouth, RI, had devised a solution which he ran in five different versions of FOCAL, BASIC, FORTRAN and PAL. I wrote Geoff and asked him to devise a possible test based on this problem solution. He did so, and also sent a sheaf of expository notes on languages, EAE (hardware extended arithmetic element), weird features, and the like. Some of this appears in this issue, although most of it will be in the next issue.

I *think* we now have an algorithm/program (hopefully only the first of many) that can be run on a wide variety of machines in virtually any language that can really compare an 8080 MPU with PTCo BASIC to a PDP-8/E with Educomp 250 BASIC to a CDC 6600 with FORTRAN IV. But let us have your thoughts too. Remember, we're talking about potential execution speed differences of 10,000 to 1!! Remember too that an algorithm to compute, say SIN (X)/ SQR (X) 100,000 times is really not nearly as useful as an algorithm that uses a wide variety of statements, nesting, recursion, etc. and is long and complex enough to exercise the interpreter or compiler efficiency.

Since all computers are tens of thousands of orders of magnitude faster than a human, does it *really* matter how they compare to each other? And isn't price/performance ratio the crucial variable anyway? Stay tuned next issue and find out! —DHA

Review of Five Small Interpreters

Steve North

As someone around here keeps saying, "A computer without software might as well be a boat anchor" (—from Nautical Chinese Proverbs of Hing Yang Ahl). Whether or not you think computers have anything to do with boat anchors, the point is well taken—hardware is only half a computer system. It is possible to invest a good fraction of the total cost of a system in software alone. So what's the best software for you? Obviously you must take your own applications into consideration, but still, some interpreters are better buys than others. Since it isn't practical to try out every interpreter on the market, *Creative Computing* will start a series of reviews on languages for 8080-based microcomputers. If you want, we could also review 6800, 6502, Z-80, and F-8 software. Our reason for starting with 8080 software is that there are simply more products being offered and more claims being made in the 8080 market.

This issue we'll start with five small interpreters. Two of these can be had simply for the cost of reproduction, so it isn't really fair to compare them with products being offered for hundreds of dollars, but we'll do it anyway, to show that there is some very good software available for almost free. The five interpreters we're starting with are CASUAL, Palo Alto Tiny BASIC, Cromemco Control BASIC, Processor Technology 5K BASIC, and MITS 4K BASIC 4.0. Next time

we'll get to some of the bigger full-feature BASICs.

Remember in using these reviews to take your own needs into consideration. Do you really care about computational speed if you just want to use your home computer for games? Probably not, since most simple games are I/O bound. Do you want a fancy text editor with your BASIC? Is it worth hundreds of dollars to you? do you want to use external data files in your application? Do you plan to try to control external devices with your computer? Or do you plan to use your computer solely for software development? We can collect facts, but in the end, it's your money....

A Review of the Review

Author, Price and Availability, and Size: Self-explanatory.
Reliability: Bugs we've found in our use of the software.
Documentation: There are two types to look for—operating instructions, and source listings of the inner works of the language. The first type is essential; the second is of use to you only if you understand machine code and wish to hack around with the language, modify it, or just understand how it works.
Speed: We used a test program that finds two whole numbers such that the two numbers and their product contain all the decimal digits (0-9) with no repetitions. For more details on the test program see Geoffrey Chase's article right after this, and page 66 of the May-June 1976 issue of *Creative.* Since interpreters that use 15-bit numbers (and a sign bit) for their math overflow when the step for checking 34 is reached, we'll run the test to 34, and up to 55 on the interpreters with floating-point math. If readers send in their own test programs we will consider using them. In the next installment of this review, we'll use another test program to check the speed and accuracy of transcendental functions.
Features: We'll list and sometimes describe the features of each interpreter. Of course we can't explain everything (the MITS BASIC manual alone is longer than a single issue of *Creative*), so make the following assumptions for the sake of simplicity:
(1) All the interpreters have a line editor (which permits one to enter, change, or delete lines by line number), character backspace, and line-delete functions.
(2) All the interpreters have a "Direct" mode of execution. If you type a statement without a line number, it is done immediately. If you typed

PRINT A,B,C

the computer would immediately respond by typing the values of A,B and C. This can be a handy debugging aid.
(3) All the interpreters permit you to break a program by typing a character (usually control-C) at the keyboard.
(4) Multiple statements are permitted on the same line. (Example: 100 PRINT A: LET Q=23: GOTO 370)
(5) A full set of six relational operations.
And finally, our User Comment.

CASUAL

Author: Bob Van Valzah
Size: 1.6K
Price and Availability: A paper tape can be obtained from the CACHE Software Library, Lloyd Smith, 530 Pierce Avenue, Dyer, IN 46311
Reliability: No bugs.
Documentation: An almost complete copy of the instructions and a source listing of CASUAL can be found in *Dr. Dobb's Journal,* Vol. 1, No. 10. The remainder is in *DDJ* Vol. 2 No. 2.*
Speed: 40 minutes to Step 34. (Stopped by operator—CASUAL does not check for overflow in arithmetic operations.)
Features: CASUAL is not BASIC, although it is BASIC-like. Instead of using keywords such as PRINT or GOTO, CASUAL uses symbols such as '.', '$', and '?'. There is, however, a close correspondence between what CASUAL can do and what BASIC can do.
Commands: NEW, LIST, RUN, O (jumps to Operating System,) TAPE and SAVE (for loading programs from mass-storage devices.)
Statements: There are only three types of statements in CASUAL:
(1) ?expression (same as BASIC PRINT expression)
(2) leftside = rightside (same function as BASIC LET, INPUT, GOTO, GOSUB, RETURN, IF/THEN, DIM, and almost everything else in BASIC.)
(3) expression (string input statement)
You may at this point be wondering how CASUAL uses an equivalency for all those things. GOTO and IF/THEN, instance, are implemented with a '.=' structure. '.' is always the value of the next line to be executed..=500 is the same as a BASIC GOTO 500..= -1 will cause CASUAL to return to the line editor (similar to BASIC STOP or END). .=0 will fall through to the next statement. To write IF X=23 THEN 400, you'd code .=400*(X=23), which will evaluate to either 0 or 400. There are other symbols in CASUAL for calling subroutines, peeking and poking memory, direct I/O to any port, and other interesting things.
Variables: A-Z, 15-bit signed integers (0 - ± 32767)
Functions: +, -, *, /
User-defined functions: The '↑' symbol is the user-defined function. It can be defined more than once in a program. It has no argument, so any variables used in it are common to the whole program.
Arrays: There are two arrays in CASUAL: a single-byte array (for value between 0 and 255) and a double-byte array (for regular CASUAL numbers.) The arrays are one-dimensional and are referenced with "(subscript) for the double-byte array, and '(expression) for the single-byte array. Before using the arrays you must define their starting addresses in abolute memory.
Machine-language subroutine interfacing: The machine-language subroutine whose address is found at location 3 is called when the '@' symbol is found on the righthand side of an expression. Parameters can be passed both ways.
Character strings: In a rudimentary way. When inputting a string, it is placed character by character in the single-byte array. It can then be processed numerically, or output using '>', CASUAL's equivalent of the BASIC CHR function.
Formatted print: None.
Editing function: None.
External files: None.
Error messages: Seven, indicated by number.
Extra stuff: Version .186 has TAPE and SAVE drivers for Tarbell cassettes. A CASUAL-VDM DRIVER is in the works. There are many features of CASUAL we haven't mentioned here—the best way to find out about them is to get a copy. CASUAL has an initializaton routine similar to MITS' which permits you to delete some CASUAL features in return for more memory. CASUAL has no subroutine nesting and does not have < = or > = relational operators.
User Comment: It's nice (finally!) to see some non-BASIC languages for microcomputers. While CASUAL is a minimum language in some respects, and it may take some

---

*In case you're a hermit, *Dr. Dobb's Journal of Computer Calisthenics and Orthodontia* is an excellent source of free software and programming hints. It concentrates mostly on low-level stuff. One issue is $1.50; one year (ten issues) is $12. *Dr. Dobb's Journal,* Box E, Menlo Park, CA 94025.

practice to use it effectively, it certainly does squeeze a lot of performance out of 1.6K of memory! Anyway, CASUAL is fun to use.

Palo Alto Tiny BASIC

Author: Dr. Li Chen Wang
Price and Availability: Paper tape can be obtained for $4, from Community Computer Center, 1919 Menalto Avenue, Menlo Park, CA 94025
Size: 2.3K
Reliability: No problems.
Documentation: A complete set of instructions and source listing appeared in *Dr. Dobb's Journal,* Vol. 1, No. 5. Vol. 2, No. 2 contained extensions for simple character strings, calling machine-language subroutines, and PEEK and POKE-ing memory.
Speed: 31 minutes to Step 34.
Features:
Commands: RUN, LIST, NEW
Statements: FOR/NEXT, GOSUB, GOTO, IF/THEN (followed by any statement), INPUT (with optional prompt string), LET, REM, RETURN, STOP
Variables: A-Z, 15-bit signed integers (0 to ± 32767)
Functions: +. -, *, /, ABS, RND (returns a value between 1 and the argument), and SIZE (returns the number of free bytes in the system).
User-defined functions: None.
Arrays: PATB has one array, named '@'. It has a single subscript and uses all available memory (so it is not dimensioned).
Machine-language subroutine interfacing: None in the original PATB.
Character strings: None in the original PATB.
Formatted print: Printing '#integer' sets the number of spaces in which an expression is printed.
Editing features: None.
External files: None.
Error messages: Three: WHAT? in response to a syntax error; HOW? in response to overflow, a missing line number, etc.; and SORRY when there is insufficient memory.
Extra stuff: Typing control-O stops all output until another control-O is entered, which is handy for entering paper tapes. PATB statements can be abbreviated: G. for GOTO, GOS. for GOSUB, E. for FOR, and so on.
User Comment: Palo Alto Tiny BASIC is an elegantly simple Tiny BASIC. It is the kind of Tiny BASIC you would expect to find on a big computer (if big computers had Tiny BASIC)—there's nothing you can do to bomb the system! PATB is powerful enough to run Star Trek in just 8K of memory (including PATB itself)! Far out! The price seems reasonable enough.

Cromemco Control BASIC

Price and Availability: $15 for a paper tape. from Cromemco, 2432 Charleston Rd., Mountain View, CA 94043.
Size: 3K, located at E400 hex.
Reliability: No problems.
Documentation. Very complete instructions are provided, but no source code.
Speed: 40 minutes to Step 34
Features: Same as Palo Alto Tiny BASIC, with the following additions:
Commands: LOCK (allocates area in memory for program files), SAVE (saves files in locked area), EPROM (burns in a file on an EPROM with a Cromemco Bytesaver), LOAD (loads file—opposite of SAVE), WIDTH (sets terminal width), NULL (number of nulls after a CR/LF), QUIT (jumps to O/S).
Statements: PUT (stores a value in absolute memory), OUT (outputs to an I/O port).
Variables: AO-ZO. Hex literals are denoted by a leading % sign, as in %F5AB.
Functions: +, -, *, /, SGN, AND, OR, XOR, GET (gets a value from absolute memory), IN (input from an I/O port), LOC (to find absolute address of array elements), CALL.
User-defined functions: None.
Arrays: Has one other array in addition to @(X), named &(X), single-byte array. It overlaps with @(X) in memory.
Machine-language subroutine interfacing: Through the CALL functions. Parameters are passed both ways through the stack.
Character strings: Strings of up to 132 characters can be accessed by $(X), the way numbers are accessed by @(X) and &(X). String and array space intentionally overlap. There is a string input statement.
Formatted statement print: Same as PATB.
Editing functions: None.
External files: No data files. As mentioned before, program files can be kept in high memory with LOCK, LOAD, SAVE, and EPROM commands. These programs can be called by other programs with a special version of the RUN command. A STOP in a called program will function as a return to the calling program.
Error messages: Same as PATB.
Extra stuff: None, other than the extensions already mentioned.
User comment: Cromemco Control BASIC is an expanded version of PATB, primarily intended for process-control applications. For this reason it seems to be more hardware-oriented, while the original PATAB does not get the user involved in knowing absolute memory locations, hex math, or I/O ports. If you need to control devices, fine, get CCB; if not, stick with PATB, which is smaller and faster. It is unfortunate that CCB is available only in copies that run at E400 (not zero), which many people will find to be a major obstacle in their use of CCB, unless they want to burn it into a few PROMs on a Bytesaver.

Processor Technology 5K BASIC

Authors: Processor Technology and Applied Computer Technology.
Price and Availability: Available on CUTS cassette for $14.50, from Processor Technology Corp., 620 Hollis St., Emeryville, CA 94608.
Size: 6.6k
Documentation: The 40-page instruction manual explains the features of the language with frequent examples. The source code is not available.
Reliability: There are no known bugs. (An earlier release of BASIC/5, called 5K BASIC, did have some problems, but these have been cleared up).
Speed: 26 minutes to Step 34; 65 minutes to Step 55.
Features:
Commands: CLEAR (clears variables), GET (loads program from cassette), EDIT (enters edit mode), LIST, NEW, RUN, XEQX (loads and executes a program).
Statements: DIM, RESTORE, READ, DATA, LET, FOR/NEXT, IF/THEN, STOP, END, GOTO, INPUT, PRINT, GO-SUB, RETURN, FILE (opens a data file), CLOSE (closes a data file), SET (to set input port, output port, speed of video display, memory size, and nul characters after a CR/LF.
Variables: A-Z,AO-Z9, floating-point numbers (.1E-127 to .999999E-127).
Functions: +, -, *, /, ABS, INT, SGN, RND, SQR, SIN, COS, TAN, TAB, ARG, CALL.

User-defined functions: None.
Arrays: Yes, one-dimensional.
Machine-language subroutine interfacing: With ARG and CALL. ARG is used to set up parameters to be passed to the machine-language program. The CALL returns a value from two of the CPU registers.
Character Strings: None.
Formatted Print: Yes, by use of %specifications% in a PRINT. This can be used to set the number of trailing zeros, number of places to the right of the decimal, exponential format, or freeform.
Editing Functions: An edit mode permits insertion and deletions of characters on a program line. Control characters (or special keys on a SOL-20) are used to control editing functions. Simple, but powerful.
External Files: BASIC/5 has provision for reading and writing data files using a Processor Tech CUTS cassette interface. The syntax required to use this feature is similar to disk file I/O in some BASICs, making it rather powerful.
Error messages: Thirteen, by two-letter error codes.
Extra stuff: 5K BASIC has a built-in VDM driver, with some interesting features: it can output at a variable rate of speed, halt entirely, turn the cursor on or off, or clear the screen.

User Comment: Processor Tech BASIC/5 is a cheap, powerful BASIC. We especially like the cassette-file I/O, which is a lot better than reading or writing data byte-by-byte. The only thing we miss is multidimensional arrays, but in view of the low price and nifty features of BASIC/5, we'll overlook that. If you don't have a CUTS cassette interface you may find BASIC/5 to be of more limited usefulness since you won't be able to use the cassette-file I/O.

MITS 4K BASIC 4.0

Author: Microsoft.
Price and availability: $150 (only $60 for owners of Altair computer, 4K RAM, and I/O module). Paper tape or MITS cassette.
Size: Something over 3K.
Reliability: No problems.
Documentation: Extensive operating instructions are provided. Unfortunately, MITS lumped the documentation for all their BASICs together, so you must scan past all the documentation on 8K BASIC and Extended BASIC. No source code is available.
Speed: 29 minutes to Step 34; 74 minutes to Step 55.
Features:
Commands: CLEAR (clears variables), LIST, NEW, RUN.
Statements: DATA, DIM, END, FOR/NEXT, GOTO, GOSUB, INPUT, LET, IF/THEN, PRINT, READ, REM, RESTORE, RETURN, STOP.
Variables: A-Z, A0-A9, floating-point numbers (1.70141E +38 to -2.9387E-38).
Functions: +, -, *, /, ABS, INT, RND, SGN, SIN, SQR, TAB, TAN, USR.
User-defined functions: No.
Arrays: Yes, one-dimensional.
Machine-language subroutine interfacing: Through the USR function. Parameters can be passed both ways.
Character strings: None.
Formatted print: None.
Editing functions: None.
External files: None.
Error messages: Twelve, by two-letter error codes.
Extra stuff: Has an initialization dialogue permitting deletion of unwanted features to save space. BASIC patches itself for MITS interface boards depending on the initial position of the sense switches. We had some difficulty using 4K BASIC with a non-MITS interface and eventually just patched out the routine which fixed up the I/O in BASIC. 4K BASIC will not run on a Z-80 CPU.

User Comment: A small, solid floating-point BASIC. The price seems rather prohibitive for people who don't own Altairs, however. If you own an Altair and the right options, and don't plan to do anything too fancy with your computer, you might consider getting 4K BASIC.

Some Notes on Languages and Another Test Installation

Geoffrey Chase

HARDWARE

16K PDP-8/E with RK05/RK8E cartridge disk, TD8/E Dectape, high-speed reader/punch for paper tape. Console terminal (there are others): LA-36 Decwriter.

The machine has the EAE hardware boards that enable a 1-by-2 multiply and a 1-into-2 divide, both integer operations, as well as certain other operations such as normalize.

The machine does not have DEC's floating-point processor hardware (FPP).

The benchmarks were taken under stand-alone OS-8.

Fortran-2

The Fortran-2 (DEC) takes no account of EAE hardware. All operations are software—simulated FPP, if you will. The floating format carries 27 bits of precision.

The "hybrid" program uses machine-language patches to speed two things only (more could be done): integer division, and subscribing.

The resulting code is nearly twice as fast. It is also shorter—the object code, that is!— even though the source code (what the programmer wrote) seems to be longer. One Fortran statement can generate many lines of object (binary) code.

The Fortran-2 system uses as a second pass the SABR relocatable assembler. Lines beginning with "S" are passed, uncompiled, to the assembler directly.

The resulting code is less than optimum. There are many, often unneeded, changes or resets of memory field (core stack); these are especially baneful under ETOS or similar systems which must interpret the virtual (user's specified) memory field into the physical (user's core partition) memory field.

There are very many "external" subroutine calls, even if the user never says CALL or defines a FUNCTION. In particular, the overhead of the subscripting routine is high.

In almost any language, *some* price is paid for the luxury of subscripts.

The Floating Fix Problem

PDP-8 Fortran-2 is limited to signed integers between -2048 and +2047. Even larger systems not infrequently allow values up to say, around 16,000; not high enough for the AHLDIG program.

Small floating values can be truncated in any Fortran by

INT = VALUE or INT = IFIX(VALUE)

VALUE = INT or VALUE = FLOAT(INT).

But floating values too large to be true integers pose a nasty problem. One solution (machine-dependent) is shown in the Fortran-2 AHLDIG.F2: create a suitable *de*-normalized 0.0 and add it to VALUE.

At the Fortran-2 programming level, without recourse to assembler language, one could try the following:

(1) find out how many binary bits of precision are carried in your floating-point ("REAL") numbers. These are often termed, rather inaccurately, "mantissa" bits. The sign bit (+ or -) doesn't count;

(2) early in the program and before repeated loops, define M = number of "mantissa" bits, and define DENORM = 2.0** (M-1).

Then, where the present Fortran-2 source code does a QUOT = QUOT + ZERO, write instead QUOT = (QUOT + DENORM) - DENORM. The parentheses are usually necessary.

This is slower than the highly machine-dependent "ZERO" code shown, but it should do the job.

Fortran is pretty rigid. There is no equivalent to BASIC's INT ( ), nor is recursion allowed in subroutines. About its only virtue—at the Fortran-2 level—is speed, and the ability to insert binary code.

Note: Strictly "classic" Fortran requires:

(1) all line numbers to be indented one space;

(2) all WRITE formats to start with a carriage-control character, for example 1H (space), before all else in the FORMAT.

(3) device numbers ("WRITE(1,200)...") differ wildly from one system to another.

Fortran-4

The DEC Fortran is much more sophisticated compiler than the Fortran-2. It does *not* allow—typical of Fortran-4's—inserted binary code. The output is highly optimized code suitable for assembly by the RALF assembler. Integer (fixed) values may range up to roughly 8,000,000.

The fly in the ointment: RALF assembler code for execution by FPP *hardware.* If, like this writer, you haven't this rather costly option, then the run-time system must *interpret* the RALF code. So we come full circle, from compilation to interpretation: from speed to slowness.

The run-time system does sense the presence or absence of the EAE multiply/divide, etc., hardware. My timings on Fortran-4 are thus very much slower than would be obtained on an FPP machine, but faster than would be obtained on a machine without EAE hardware.

Educomp OS-8 BASIC

This is a compiled language with an interpretive run-time system that makes some use of EAE hardware if present. Not surprisingly, the run-time is not too much slower than Fortran-4's (less than twice the run time). If, like PDP-11 RSTS, a true integer mode were available, the two languages might wind up in a near-tie.

Edu-20 BASIC (DEC)

This is maybe a "worst case" for purely interpretive languages. FOCAL and the Educomp EDU-200/500 would almost certainly be faster.

Still, pure interpreters are instrinsically slow when number-crunching is required. All will show a very marked difference in time relative to any of the preceding languages.

(It might be mentioned that FOCAL—not shown here—has been considerably expanded and to some extent speeded by non-DEC programmers. J. van Zee at the Chemistry Dept., University of Washington, Seattle, currently markets a "U/WFOCAL" that is a large superset of DEC FOCAL and runs faster.)

Summary of Timings

	TO '55'		TO END		
	Hr	*Min*	*Hr*	*Min*	
Hybrid		2.2		4.9	
Fortran-2		4.1		9.4	
Fortran-4		10.1		23.9	
Educomp Basic		17.3		40.8	
Edu-20 Basic	5	44.8	13	35.0	est.

Hybrid by author; Edu-Basic from Educomp (Hartford); Others from DEC (Maynard)

Program Listing: Fortran-4

```
      C    "AHLDIG.F4"      PAS      3/77

           INTEGER DIGIT(10),CHAR(3),A,B,C,A1,A2,QUOT

           DATA CHAR /'A','B','C'/

           DO 490 A = 12,98
           WRITE (4,100)A
           A1 = A/10
           A2 = A - 10*A1
           IF (A2 .LT. 2) GO TO 490
           IF (A2 .EQ. A1) GO TO 490
           MODA = MOD(A,2)

      C    ...WHICH GIVES MODA = 1 IF A2 IS ODD, ELSE 0

           ITEMP = 10000/A

           DO 470 B = ITEMP,987

           DO 230 I = 1,10
      230  DIGIT(I) = 0

           DIGIT(1+A1) = 1
           DIGIT(1+A2) = 1
           QUOT = B/10
           J = B - 10*QUOT
           IF (J .LT. 2) GO TO 470
      310  MODB = MOD(B,2)
           IF (A2+MODB .EQ. 6) GO TO 470
           IF (J +MODA .EQ. 6) GO TO 470

      C    ...KNOCKS OUT 5*ODD AND 6*EVEN

           IF (DIGIT(1+J) .GT. 0) GO TO 470
           DIGIT(1+J) = 2

           ITEMP = QUOT
           QUOT = QUOT/10
           J = ITEMP - 10*QUOT
           IF (DIGIT(1+J) .GT. 0) GO TO 470
           DIGIT(1+J) = 2
           IF (DIGIT(1+QUOT) .GT. 0) GO TO 470
           DIGIT(1+QUOT) = 2
```

```
              C = A*B
              ITEMP = C

              DO 400 I = 1,4
              QUOT = ITEMP/10
              J = ITEMP - 10*QUOT
              IF (DIGIT(1+J) .GT. 0) GO TO 470
              DIGIT(1+J) = 3
              ITEMP = QUOT
      400     CONTINUE
              IF (DIGIT(1+QUOT) .GT. 0) GO TO 470
              DIGIT(1+QUOT) = 3

      C       SUCCESS:

              WRITE (4,150)A,B,C

              WRITE (4,110) (CHAR(DIGIT(I)),I = 1,10)
              WRITE (4,100)

      470     CONTINUE

      490     CONTINUE

              STOP

      100     FORMAT(1H ,1I3)
      110     FORMAT(1H ,10A1)
      150     FORMAT(1H0,2I7,F8.0)

              END
```

Program Listing: Educomp OS-8 BASIC

```
100 REM.       "AHLDIG.BA" V.4 PAS      3/77
110 DIM D(9)
120 !  ----------
130 FOR A=12 TO 98
140 PRINT A
150 A1=INT(A/10)
160 A2=A-10*A1
170 IF A2< 2 THEN 510   ! ENDS IN 0 OR 1
180 IF A2=A1 THEN 510
190 M1=A2-2*INT(A2/2)   ! 1 IF ODD, 0 IF EVEN
200 !  ======
210 FOR B=-INT(-10000/A) TO 987
220 FOR I=0 TO 9
230 D(I)=0
240 NEXT I
250 D(A1)=1
260 D(A2)=1
270 V=B
280 F=2
290 Q=INT(V/10)
300 J=V-10*Q                 ! REMAINDER=LOWEST DIGIT
310 IF J<2 THEN 490          ! 0 OR 1
320 M2=J-2*INT(J/2)          ! 1 IF ODD, 0 IF EVEN
330 IF A2+M2=6 THEN 490         ! 5*ODD OR ELSE 6*EVEN
340 IF J+M1=6 THEN 490          ! SAME CHECK
350 GOSUB 570
360 IF F<0 THEN 490
370 C=B*A
380 F=3
390 V=C
400 GOSUB 550
410 IF F<0 THEN 490
420 PRINT " A"," B"," C"
430 PRINT A,B,C
440 FOR I=0 TO 9
450 PRINT CHR$(64+D(I));
460 NEXT I
470 PRINT
480 PRINT
490 NEXT B
500 !  ======
510 NEXT A
520 STOP
530 !  ----------
540 V=Q
550 Q=INT(V/10)
560 J=V-10*Q
570 IF D(J)=0 THEN 600
580 F=-1
590 RETURN
600 D(J)=F
610 IF Q>9 THEN 540
620 IF D(Q)>0 THEN 580 ! AS BEFORE
630 D(Q)=F
640 RETURN
650 !  ----------
660 END
```

Timing Run: Educomp OS-8 BASIC

```
EXECUTE AHLDV4
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 A              B              C
 27             594            16038
CCACBBCACB

 28
 29
 30
 31
 32
 33
 34
 35
 36
 A              B              C
 36             495            17820
CCCABBACCB

 37
 38
 39
 A              B              C
 39             402            15678
BCBABCCCCA
 40
 41
 42
 43
 44
 45
 A              B              C
 45             396            17820
CCCBAABCCB

 46
 A              B              C
 46             715            32890
CBCCABABCC

 47
 48
 49
 50
 51
 52
 A              B              C
 52             367            19084
CCABCABBCC

 53
 54
 A              B              C
 54             297            16038
CCBCAACBCB

 55
 56
 57
 58
 59
 60
 61
 62
 63
 A              B              C
 63             927            58401
CCBACCABCB
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 A              B              C
 78             345            26910
CCCBBBCAAC

 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98

STOP AT LINE 520
READY

40 MINUTES, 47 SECONDS
```

The Pocket Computer is *(Almost)* Here

Richard A. Ahern*

In at least two semiconductor laboratories in this country, technical obstacles have already been overcome and production problems are now being solved on a battery-powered, pocket-sized general-purpose computer that can:

1) sequentially access up to 13 million bytes of data from a self-contained half-ounce high-data-density microcassette,

2) randomly access (or nearly so) up to 100,000 bytes of data from an internal continuous-loop tape,

3) show up to 378 characters simultaneously on a low-power 2½ × 2½-inch thin-film "flat screen" display,

4) display graphics and slow-frame-advance low-resolution video on the same 128 × 128-dot dot-matrix screen, and

5) send and receive at up to 1200 bauds.

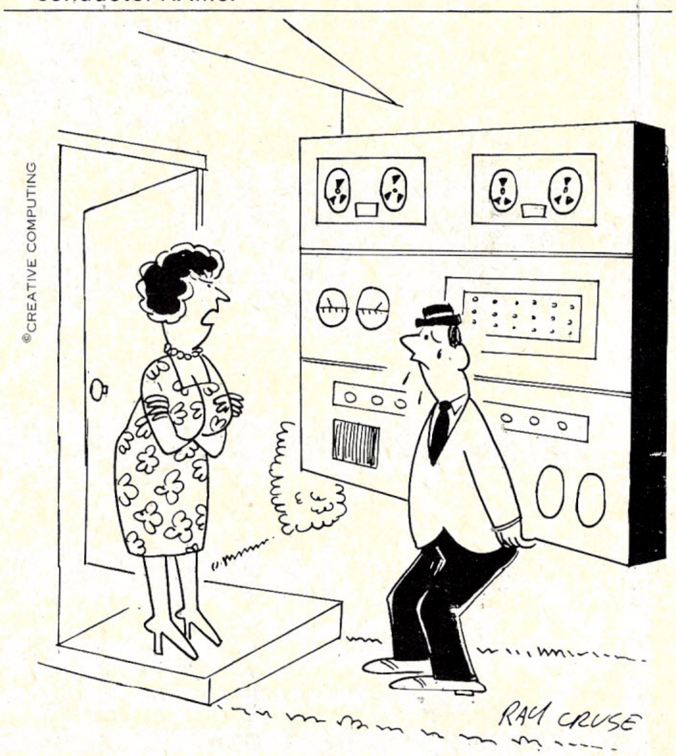
Introduction

The pocket computer will be able to do much of what many general-purpose desk-top computers now do, but will be almost as small and as inexpensive as the more advanced of the recently introduced programmable pocket calculators.

The device itself, in its most all-inclusive configuration, will look like a pocket dictating machine with a calculator keyboard and a miniature TV screen on its face. Although it will have only several more keys than many programmable calculators, it will be capable (with a shift key) of full USASCII character generation. Although its screen will resemble a small TV screen (2½ × 2½-inches, or so), it will actually be a lightweight (3 to 4-ounce), X-Y-addressable thin-film dot-matrix display (probably either 128 × 128 or 256 × 64). The long-awaited "flat screen" display was finally introduced (with production commitment) at the December, 1975, International Electron Devices Meeting.

The high-data-density microcassette subsystem, although less eagerly awaited (indeed, hardly talked of at all within the industry until its recent advent), is nevertheless the key to the power of the pocket computer. The common wisdom has long seemed to have had it that to achieve an error rate of 10^{-6} bits or better on a cassette system, there should be no more density than 800 BPI on only two tracks per cassette. However, a well-known nonprofit research organization has recently developed a production-ready cassette subsystem with 7000 BPI and 8-track-per-cassette densities that has an error rate that is still 10^{-6} bits.

Furthermore, work is now being done to develop a continuous-loop microcassette system capable of accessing up to 100,000 bytes of data at an average access time of 900 milliseconds (based on 15 inches of continous-loop tape moving at 8½ ips at the above densities). Such a system would be able to replace or supplement high-ROM-need systems and would be able to supplement semiconductor RAMs in almost any random-access system, allowing subroutines in a program to be called in (or, indeed, even entire sequential programs), one at a time, and reducing the need for semiconductor RAMs.



"Must you always bring your work home with you?"

*Applied Electronics, P.O. Box 161, Forest Hills, NY 11375

Conventional Uses

The pocket computer will be able to be used for many of the same administrative purposes that intelligent terminals and small general-purpose computers are now required for, especially order entry and other "fill in the blank" operator-prompt/data-check remote batch-processing jobs. It will be especially helpful when a user wishes to take it from place to place (or home at night) for it should weigh only 20 ounces or so, not 40 to 60 pounds as its counterparts of today do. In addition to its administrative functions, the pocket computer will be usable in many scientific applications; those that do not require extremely fast memory functions.

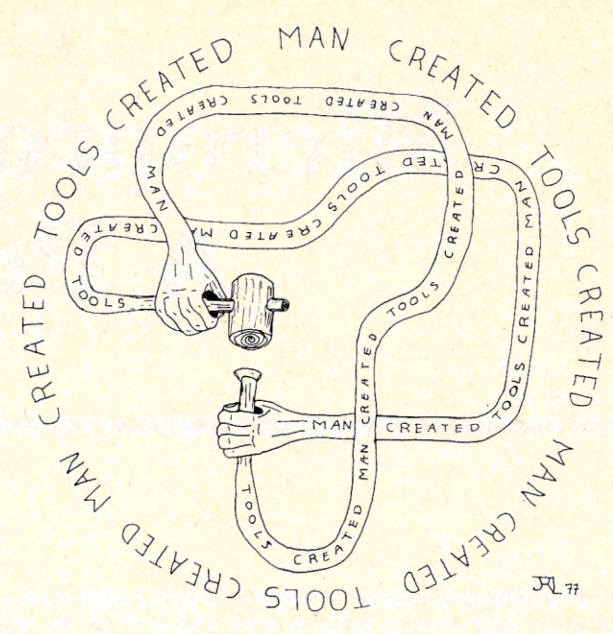
The device will probably not have a printer as part of it, but hardcopy—either selective or entire reports—will be able to be easily obtained by holding the pocket computer's "send" telecommunication port to the telephone mouthpiece after first calling any ASCII computer, printer or Teletype (speeds of 1200 bauds can easily be achieved with small microelectronic battery-powered acoustical couplers). Alternatively, the microcassette could be removed from the pocket computer and brought to any computer station with a microcassette reader. (Which we will see many of as pocket computers are introduced).

Technical Problems

Like pocket calculators, pocket computers will be designed to be able to be put together by hand in 10 to 20 minutes. Although there will be a troublesome interconnect problem—especially with display screens of 128 × 128 dots or larger—the two most significant technical problems lie in fully debugging the microcassette (especially the continuous loop) and display subsystems.

Regarding the microcassette system, reading or writing continuously at high tape densities is not a problem. Provided the address is not too long, the tape blocked by record and the tape speed kept down, searching can also easily be done. However, frequent starting and stopping followed by selective reading and writing can become more difficult (in high start/stop systems that require only data checking and indications of nonstandard conditions, analog pulses can be added to the tape). The continuous-loop microcassette, when used only as a ROM replacement or addition, should present no problems. Problems will arise when too much writing has to be done and complex housekeeping routines have to be set up. If high-density selective writing results in too high an error rate, low-density writing could be used for RAM or, in some cases a system of double addresses and an either/or search system could be used to reduce the error rate.

There are also problems with the display subsystem, but again none that are not able to be solved. The cadmium selenide electroluminescent phosphorus thin-film display which has been most discussed recently requires a voltage that is probably too high for a hand-held device. Electrochromic displays, although they add the prospect of color and a memory within the display itself, require another year or two of development. Liquid crystal displays (LCDs) react too slowly to handle rapidly-changing illumination configurations (as would be needed to handle, say, large amounts of descriptive data in an "advancing line" reading mode), normally have no shades of gray and have X-Y-addressing duty-cycle problems. The first pocket computers will probably contain LCD displays, for although they are slow to react, they are cheap and the technology is available now to mass produce them for only a little more than a 12-character line of light-emitting diodes (LEDs)



Will the circle remain unbroken?

Marketing Problems

Although the two companies that have introduced magnetic-card-reading programmable calculators consider these devices small general-purpose computers (and technically they are, according to the textbook definition of a computer: an electronic I/O device with a memory that follows a program it can store and responds to co-directional instructions), most of us would include in the definition of a computer the requirements of full alphanumeric character generation and display capabilities, a multiline display and extended sequential and random access. The companies who are most capable of introducing the pocket computer have a vested interest in the status quo, for they all also produce programmable terminals and/or small general-purpose computers and hence are reluctant to add a new competing line, and a lower-priced and lower-profit one at that. Primarily for this reason, special-purpose new-market versions of the pocket computer will probably be introduced first, the initial ones certainly before the end of the decade.

One version would be for computer-assisted instruction. Another would be for crime countermeasures, such as searching a pre-recorded microcassette (updated each morning over the telephone) for a license-plate number, a person's name, a vehicle registration number, a stock certificate, bond or currency number.

In years to come, the sky seems to be the limit. Future general-purpose versions of the pocket computer will allow numerous innovative uses, even the high-speed loading of newspapers, reference data or books at special coaxial loading stations. For more specific and less extensive data needs, eventually (with perfected optical scanning) the entire Library of Congress will be machine-readable and capable of being accessed through any telephone. ■

Microprocessors — A Primer

Theodore J. Cohen, PhD

A sophisticated electronic device known as the microprocessor will shortly have a profound impact on our way of life. Within a year or so, this device, about half the size of a matchbook, will be incorporated in a variety of consumer products ranging from automobiles to digital watches.

What are microprocessors? Why are they important? How do they function? And how will they be used in consumer products? These are the questions answered here.

The Heart of a Computer System

All computers consist of five basic sybsystems:

- An input device through which instructions and data are entered into the computer;
- A central processing unit (CPU) which controls the computer's operation;
- An arithmetic logic unit (ALU) which performs mathematical operations;
- A memory, in which instructions and information are stored;
- An output device, through which processed data leave the computer.

The heart of this basic computer system, which consists of the central processing unit (CPU) and the arithmetic logic unit (ALU), can be incorporated on a single integrated circuit (IC) chip, and this chip is known as a microprocessor.

While early microprocessor-based computers required a considerable number of IC's (30 or more) to recover data from memory, second-generation microprocessors permit the construction of computers having as few as two chips. Thus, it is not unusual to find that the microprocessor is often referred to as a "computer on a chip."

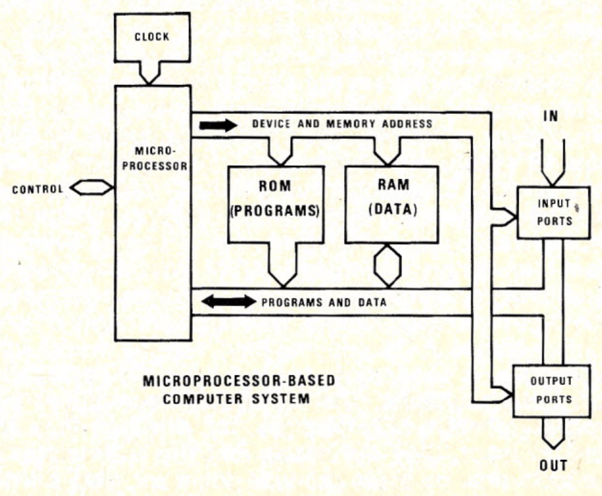
A New Electronics Era

Despite the rapid advances which have been made in electronics since the introduction of the transistor some 30 years ago, many consider the development of the microprocessor as heralding the beginning of a new electronic era. The reasons for this are many. For example, some arithmetic and computational capabilities available in today's microprocessor-based systems would be impractical to duplicate using more conventional circuitry. Then, too, the use of microprocessors results in drastically-reduced product design time, reduced product complexity, and hence, lower product cost. Finally, microprocessor-based products can be programmed to execute a sequence of instructions, and thus can control, or interact with, a variety of instruments, machines, and systems.

In short, the capabilities inherent in microprocessor-based products represent a significant advance in computational and control circuit design.

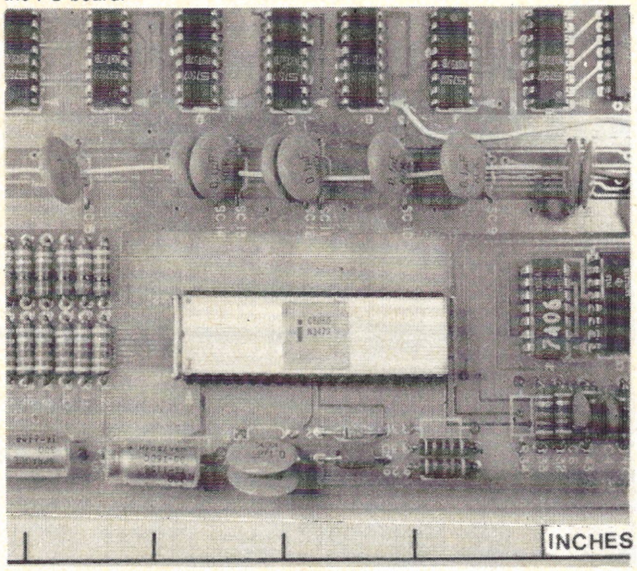
The Microprocessor as a Circuit Element

As already seen, a microprocessor can form the heart of a computing system . . . the heart of a microcomputer, if you will. Here, the microprocessor, together with such additional components as read-only memories (or ROM's, which are used to store the microcomputer program), random-access memories (or RAM's which are used to store data) and interfaces for peripheral devices, is so connected as to perform computations and to make decisions. The microprocessor determines what external devices should provide or have access to data, performs calculations using the data provided, and makes decisions based on these calculations and upon timing constraints which may be imposed by the user. Looked at another way, the microprocessor, which is only one component of a microcomputer, coordinates the activities of the memories and the input-output devices, and



MICROPROCESSOR-BASED COMPUTER SYSTEM

Intel's 8080 microprocessor, a popular second-generation device, contains the equivalent of 5000 transistors as well as most of the basic operational features found in present-day minicomputers. The chip itself is about 1 cm^2 and 0.1 mm thick. It is mounted on a plastic package called a DIP (Dual In-Line Package) about 5 x 1-1/2 cm. The MPU is dwarfed by the other discrete components (resistors and capacitors) on the PC board.

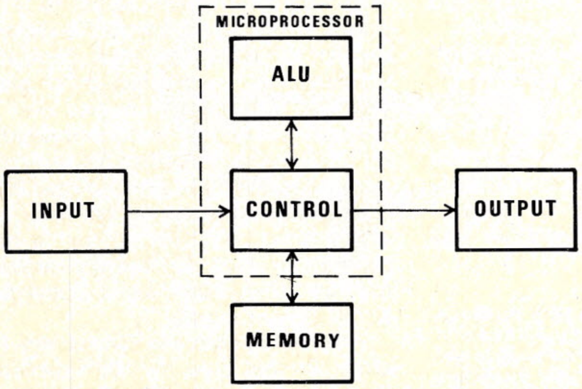


performs logical or arithmetic operations on the data stored in RAM. Used in this manner, the microprocessor makes it possible to incorporate decision-making and data-processing capabilities in a variety of products ranging from automobiles to watches, and from calculators to television receivers.

There's a Microprocessor in Your Future

If you drive an automobile — and most of us do — there's a microprocessor in your future. The need for more dependable, fuel-efficient vehicles makes the automobile a prime candidate for early applications of microprocessor technology. Through the use of an on-board microcomputer, it will soon be possible to monitor such diverse parameters as engine speed, ignition timing, engine temperature, compression, and emission, and to determine automatically that point where fuel economy and emission control are optimized. It will also be possible to determine more accurately when shifting should occur, thereby minimizing transmission damage. Even diagnostic analyses of critical engine functions will give the driver advance warning of impending breakdowns.

While the on-board microcomputer is monitoring your vehicle's performance, it will also be watching out for you, making your ride smoother and safer. Don't worry about your doorlocks; the computer will lock the doors for you once your car's speed exceeds 5 m.p.h. The microprocessor-based computer will also monitor your braking system (to prevent lock-up), and your speed (to warn of excesses). The onboard computer will even be able to provide anti-theft security by disabling the ignition control system when your car is entered without a key having been used.



A BASIC COMPUTER SYSTEM

Now that your appetite is whetted, consider how microprocessors will be used to improve the performance and capabilities of the following products:

Digital Watches

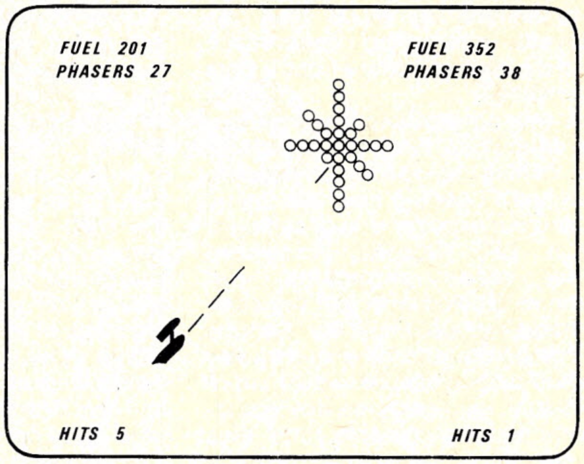
Engineers are already working on microprocessor-based watches that will include a calculator, an alarm, and an interval timer. It may even be possible, someday, to own a watch which provides personal physiological monitoring.

Hand-Held Calculators

While mature, the calculator market is certainly not saturated. Newer more complex units will soon be available, and some may even be able to monitor such body functions as blood pressure or pulse count.

Television Receivers

Microprocessors will permit the expansion of today's television receiver into a comprehensive recreational and entertainment center. Through advanced technology, it will be possible to play a wide variety of games, either against an opponent or against the microcomputer itself. Liberal use of color will make video games more exciting as will the generation of more realistic game sounds. It will even be possible to play games which provide a challenge to players having a wide range of skills; in this way, users will not lose interest as their skills improve.



An Electronic Revolution

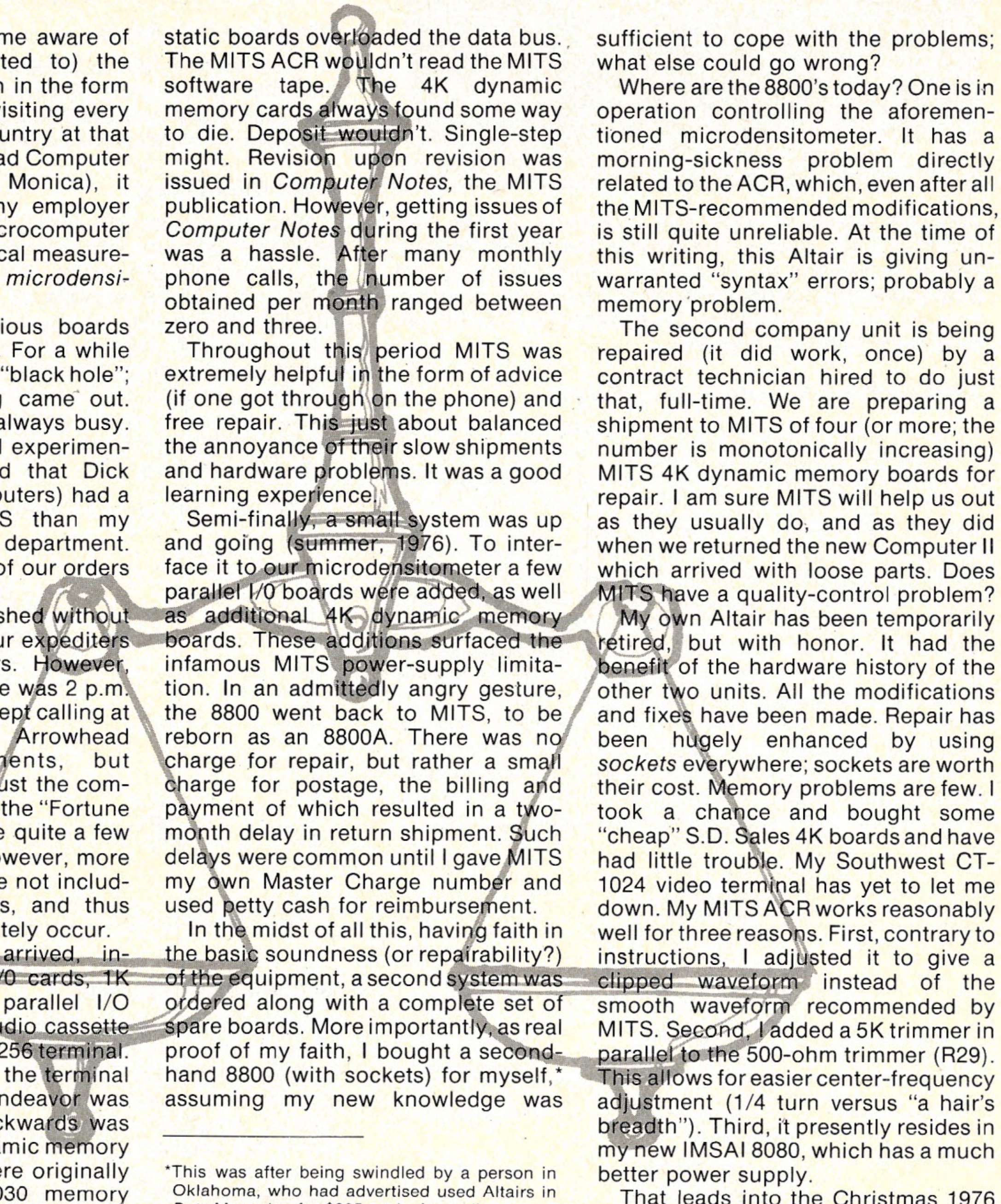
A revolution is upon us! Developments in microprocessors are changing the electronics industry at an unprecedented rate. As a result of the changes, a new generation of "smart" consumer products will soon be available . . . products which are not only more capable, but which are also designed to analyze data and to make decisions which permit a variety of tasks to be performed in a highly efficient and dependable manner.

Portions of this article appeared in "Microprocessor Technology — An Electronic Revolution," T. J. Cohen, *Sea Technology,* March 1976. ■

A Creative Computing Equipment Comparison. . . .

A Microcomputer Tale

F. R. Ruckdeschel



Two years ago I became aware of (and immediately addicted to) the microcomputer revolution in the form of the Altair 8800. After visiting every computer store in the country at that time (that is, the Arrowhead Computer Co. on Pico in Santa Monica), it became apparent that my employer desperately needed a microcomputer system to control an optical measurement device called a *microdensitometer.*

A mainframe and various boards were ordered from MITS. For a while MITS appeared to act as a "black hole"; orders went in, nothing came out. Worse, their phone was always busy. Through some controlled experimentation it was determined that Dick Heiser (Arrowhead Computers) had a better channel to MITS than my employer's purchasing department. Thus we switched many of our orders to Arrowhead.

This was not accomplished without some difficulties. First, our expediters enjoy talking to vendors. However, Arrowhead's opening time was 2 p.m. EST, and our expediters kept calling at 11 a.m. EST. Second, Arrowhead reliably made shipments, but sometimes labeled with just the company name (we are one of the "Fortune 500" companies and have quite a few thousand employees). However, more importantly, invoices were not included with many shipments, and thus payment did not immediately occur.

Eventually hardware arrived, including a mainframe, SI/O cards, 1K static memory boards, parallel I/O boards, an MITS ACR (audio cassette interface) and a Compter 256 terminal. The 1K static boards and the terminal indicate how early this endeavor was started. Later a step backwards was taken when MITS 4K dynamic memory boards arrived. These were originally populated with TMS 4030 memory chips; we have been attempting to live with those boards ever since.

A year of problems followed. The 1K static boards overloaded the data bus. The MITS ACR wouldn't read the MITS software tape. The 4K dynamic memory cards always found some way to die. Deposit wouldn't. Single-step might. Revision upon revision was issued in *Computer Notes,* the MITS publication. However, getting issues of *Computer Notes* during the first year was a hassle. After many monthly phone calls, the number of issues obtained per month ranged between zero and three.

Throughout this period MITS was extremely helpful in the form of advice (if one got through on the phone) and free repair. This just about balanced the annoyance of their slow shipments and hardware problems. It was a good learning experience.

Semi-finally, a small system was up and going (summer, 1976). To interface it to our microdensitometer a few parallel I/O boards were added, as well as additional 4K dynamic memory boards. These additions surfaced the infamous MITS power-supply limitation. In an admittedly angry gesture, the 8800 went back to MITS, to be reborn as an 8800A. There was no charge for repair, but rather a small charge for postage, the billing and payment of which resulted in a two-month delay in return shipment. Such delays were common until I gave MITS my own Master Charge number and used petty cash for reimbursement.

In the midst of all this, having faith in the basic soundness (or repairability?) of the equipment, a second system was ordered along with a complete set of spare boards. More importantly, as real proof of my faith, I bought a second-hand 8800 (with sockets) for myself,* assuming my new knowledge was sufficient to cope with the problems; what else could go wrong?

Where are the 8800's today? One is in operation controlling the aforementioned microdensitometer. It has a morning-sickness problem directly related to the ACR, which, even after all the MITS-recommended modifications, is still quite unreliable. At the time of this writing, this Altair is giving unwarranted "syntax" errors; probably a memory problem.

The second company unit is being repaired (it did work, once) by a contract technician hired to do just that, full-time. We are preparing a shipment to MITS of four (or more; the number is monotonically increasing) MITS 4K dynamic memory boards for repair. I am sure MITS will help us out as they usually do, and as they did when we returned the new Computer II which arrived with loose parts. Does MITS have a quality-control problem?

My own Altair has been temporarily retired, but with honor. It had the benefit of the hardware history of the other two units. All the modifications and fixes have been made. Repair has been hugely enhanced by using *sockets* everywhere; sockets are worth their cost. Memory problems are few. I took a chance and bought some "cheap" S.D. Sales 4K boards and have had little trouble. My Southwest CT-1024 video terminal has yet to let me down. My MITS ACR works reasonably well for three reasons. First, contrary to instructions, I adjusted it to give a clipped waveform instead of the smooth waveform recommended by MITS. Second, I added a 5K trimmer in parallel to the 500-ohm trimmer (R29). This allows for easier center-frequency adjustment (1/4 turn versus "a hair's breadth"). Third, it presently resides in my new IMSAI 8080, which has a much better power supply.

That leads into the Christmas 1976 story. By that time, my 100% record for visiting computer stores had dropped to about 75%; there were many more

*This was after being swindled by a person in Oklahoma, who had advertised used Altairs in *Byte* Magazine for $285 each. According to one of many letters from various Postal Inspectors (which is a story in itself), the culprit was caught. But where's my money?

stores. However, this batting average included the basements and garages in upstate New York, all the "real" stores in New York, Toronto, Virginia and Florida, as well as all the superb stores in California. In those travels two new and handsome machines were very visible; the Altair 8800B and the IMSAI 8080. There was no question that these machines were a step up from the 8800A, and I was determined to obtain one.

Four general features led me to getting the IMSAI. First, after actually getting blisters toggling front-panel switches on the 8800A, the IMSAI's *wide* paddle switches were very inviting. Second, the IMSAI kit offered by the local computer "store" (Memory Merchants, somewhere in Spencerport, somewhere in New York State) was less expensive than the corresponding 8800B. Fourth, and very important, two-week delivery was assured by the IMSAI dealer while, by experience, three months appeared to be more characteristic of MITS.

The IMSAI arrived on schedule. I quickly assembled it *with sockets,* and it didn't work. Using a wonderful little tool called a logic probe (from E&L Instruments), a bad IC was located on the front panel (which is a pain to mount and remove). The recommended "protect modification" was added, and the unit has in general worked well since. However, there are some subtle glitches associated with using memory boards other than those produced by IMSAI.

At about the time I obtained my own IMSAI (Jan. '77), two additional company orders were placed; one for an Altair 8800B and one for an IMSAI 8080. As expected, the Altair has yet (mid-May) to arrive. The rumor I heard was that there was a delay related to a power-supply problem. The assembled IMSAI arrived within a month. However it didn't work. Our technician found a potential problem with clock noise near the clock-generator chip, but that was not the source of *the* problem. The unit was returned to our local IMSAI dealer who, one weekend later, returned the machine with a simple explanation. The MITS ACR we were using was killing the SINP and SOUT signals. Guess what is being returned.

The present state of this story is that my IMSAI 8080 and Altair 8800 are both working, though the IMSAI is preferred. My employer has a working Altair 8800A which is often left on overnight because of the MITS ACR tapereading problem (which is common to the four ACR's the company and I have). The remaining IMSAI and Altair are operational, but not in use.

My personal evaluation, based on limited experience, of IMSAI and MITS products is summarized in the box.

Opinionated Comparison of IMSAI/MITS

	MITS	IMSAI
Delivery Time	*Very* slow	Relatively fast
Cost	8800: low enough to be enticing	Expensive, but on the whole worth it
	8800B: expensive for what you get in comparison to IMSAI	
Design Reliability	1K static boards: OK	
	4K dynamic boards: bad	?
	ACR: awful	?
	CPU card: needed modification	May need mod.
	Front Panel: 8800A hard on the fingers	Pretty and functional
	Power Supply: not good in 8800	Great
Factory and Dealer Help		
Cooperation	Excellent	Excellent
Expertise	Very good	Dealer appears more knowledgeable than factory.
Software	Very good, when we get it loaded	Not much available; rumor has it that there are problems.

My general impression is that IMSAI provides a better purchased piece of hardware, but the hardware debug and software support may not be as good as MITS. MITS should be congratulated on starting the microcomputer revolution and providing a complete software and hardware support organization. However, I would expect IMSAI to become a formidable competitor.

A final point. Already my CT-1024 is obsolete. Such hardware should become technically obsolescent within a year. If not, the technology is not progressing fast enough. However, what shouldn't happen is for the device to become outdated between the time the order is placed and the time the item is received. MITS's slow delivery borders on this time window which, for me, is a *very* negative feature of their service.

Biography

Fred Ruckdeschel is an unwilling native of New York State, presently employed as a Principal Scientist by the Xerox Corporation. He has been involved with computer programming dating back to the IBM 650 and 1401, and more recently has performed physical simulations on the IBM 360 and Sigma 7/9. With the publication of the Altair article in *Popular Electronics,* he became interested in computer hardware and has been soldering ever since.

Pub. Note: The state of the new user of microcomputers, particularly hobbyists, at this point, is that the main interest is in getting a system (read hardware) up and running. In the long run, however, attention will focus on the use of the system, with software becoming of crucial importance. And, as you mention, MITS software is outstanding. MITS 4K, 8K and 12K Basic, as well as the first packages from ASDC (Altair Software Distribution Company) are very impressive, hence as a total system (hardware plus software) the Altair is a formidable contender.

Pertaining to delivery (and not to make excuses for MITS), DEC is by far and away the largest and most successful vendor of minicomputers, yet traditionally DEC has had the worst delivery record. Today, you can get an 8A faster from an OEM (equivalent of store) than you can from DEC (nine months delivery yet it's been in production for nearly three years).

Is there a message in all this? Yes. Shop cautiously and wisely to meet your own needs, and caveat emptor. — DHA. ■

Radio Shack's $600 Home Computer

Wes Thomas

This September you can walk into any of the 140 major Radio Shack stores around the country and buy a fully wired and tested TRS-80 microcomputer for $599.95. For that amount, you get a compact ASCII keyboard with built-in microcomputer, plus a video monitor, and an audio-cassette recorder to store programs or data. You can take all of these home, plug them in, and write your first program in Radio Shack's "Level I" BASIC. Or you may decide to buy just the microcomputer/keyboard for $399.95, and use your own TV (through an RF modulator) along with your own home audio-cassette recorder.

Radio Shack has thoughtfully provided five groups of programs for the TRS-80 on audio cassettes: "Blackjack and Backgammon" (free), "Payroll" (for up to 15 people, at $19.95), Educator—Math I" ($19.95), "Kitchen" (menus, conversion tables, computer directory and message center, for $4.95), and "Personal Finance" ($14.95).

I had a chance to try out the TRS-80 at Radio Shack's press conference in New York on August 3. I found it easy to use, with a compact, comfortable keyboard, a legible 12-inch video display, and a well-written, thorough, and very patient instruction manual. The video display features automatic scrolling, and displays 16 lines of 64 characters for text (software-selectable to 32 characters per line).

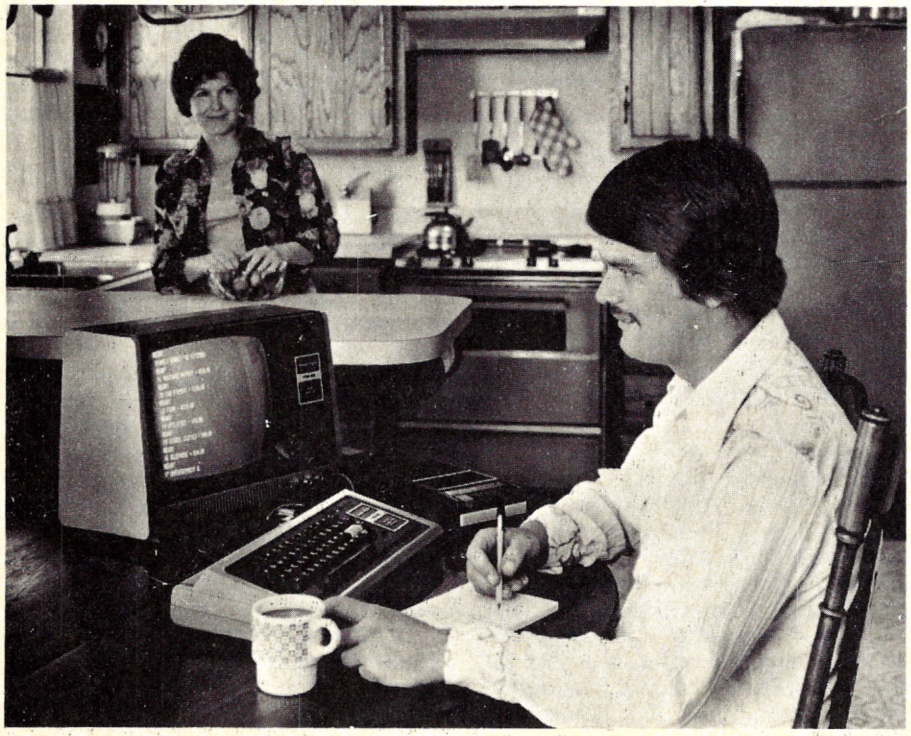
Father of the TRS-80

The TRS-80 is the brainchild of 25-year-old Steve Leininger, who spent two years designing National Semiconductor's SC/MP Development System. Steve was originally hired by Radio Shack last year to develop a computer kit, but, having worked in a Byte Shop in California, he quickly convinced Radio Shack that "too many people can't solder." Steve says he put together the TRS-80 based on software ideas from *Creative Computing* and other sources.

Enhanced Tiny Basic

The TRS-80 uses Radio Shack's "Level I" enhanced Tiny BASIC interpreter, which resides in 4K of ROM (Read-Only Memory). Level I improves on Tiny BASIC by adding floating point (to provide the decimal point), and scientific notation. And it adds graphics commands that let you easily generate pictures, which you can mix in with words. The storage capacity for programs plus data is 4K (about 500 characters) in dynamic RAM (read and write memory). When you run out of space in internal computer memory, you can use the cassette recorder to extend it. (Note to hardware fans: Radio Shack is using the Z-80 chip for the TRS-80, but not the S-100 bus. And they're using "Radio Shack" cassette-tape protocol, which adds one more to the dozen tape formats already in existence.)

Incidentally, the microcomputer is designed and manufactured by Radio Shack in the U.S., the video monitor is from Taiwan, and the cassette recorder is Japanese.



Peripherals

Is that all there is? Nope. Radio Shack has bigger plans for December. Here is what you can get for Christmas, if you can convince your wife (or husband or whoever) that you desperately need a computer to make

your first million (Radio Shack says they will pay for useful programs):

- A compact 80K floppy-disk unit for $600-$700 (including disk operating system).
- A dot-matrix printer for under $1,500 (a $700-or-less printer is in the works for 1978).
- Extended (8K) BASIC, or "Level II," as Radio Shack calls it (this includes extended string capability, and other enhancements that will allow more sophisticated business and educational programming).
- Level III BASIC (with disc-control commands, and assembly-language subroutines). An internal 12K ROM will be available for this, allowing 4K for other uses.
- Assembler (resident in 16K of disk or on cassette). This will permit faster graphics and I/O routines and let you do a lot of fancy tricks.
- FORTRAN IV (in 16K of disc).
- New software packages for small business and education, including general ledger, accounts receivable, inventory control, music theory, long division, algebra, and other programmed-instruction packages now being developed by noted computer-aided-instruction experts.
- Text editor (using disk and 8K of RAM).

Radio Shack is also planning a modem (non RS-232) so you can connect your computer to other home computers or to a time-sharing service and let it talk its own language. Or you can use it like an intelligent terminal for business-data entry. Or send "electronic mail" messages. Or whatever else you have in mind.

Also in the planning stages is an "Expansion Unit" to hold extra RAM memory, up to a total of 62K. Or you can put up to 16K of that inside the keyboard case (the TRS-80 with 16K of RAM is an extra $289) and another 16K inside the disk unit (ROM memory can be expanded in the keyboard case up to 12K). Now you're ready to go into the business of serving small business — and to make your first million.

Future Plans

Radio Shack is also thinking about other features, and intends to provide them if enough customers are interested, such as:

- Color graphics and an expanded graphics instruction set, plus an RF modulator (pending FCC approval) so you can hook up your color TV and invent your own interactive video games and impress your friends.
- An External Device Controller to interface analog signals, such as from a joystick for video games, or from sensors for fire detection, etc. (a convenient 40-pin plug on is located the back of the keyboard unit).
- Music and speech synthesizer, and voice-recognition devices.

Markets

Radio Shack is going into the computer business in a big way. Eventually, 2100 stores will carry the TRS-80. The main target market is small businesses, then education, followed by the consumer market. Radio Shack's principal objective, according to their financial consultant Garland Asher, is to "influence other product lines — like digital scanners — to improve Radio Shack's image, and to move the corporation into higher-ticket merchandise."

Meanwhile, Radio Shack's parent, Tandy Corp., is planning a chain of "Tandy Computer" retail stores across the country; the first one will open Oct. 1 in Fort Worth. According to VP John Gatliff, "We'll carry a broad line of other vendors' items too, and we'll also sell by mail order, starting in October. Our markets are business, education, and hobbyists."

As Radio Shack president Lewis Kornfeld put it, "This device is inevitably in the future of everyone in the civilized world — in some way — now and for as far ahead as one can think."

Quick Comparison with Heathkit H8 and PET Computers

How does the TRS-80 compare with two other recently-introduced low-cost home computers?

- *Heath H8 kit:* $375. Adding the video-terminal kit plus cassette interface plus 8K memory brings the cost up to $1,110.
- *Commodore PET:* About the same price ($595) as the Radio Shack TRS-80. Compact and with 8K BASIC. Drawbacks: a small calculator-style keyboard, small 9-inch video screen, less-powerful MPU chip (6502), and uses the IEEE bus (which requires intelligent peripherals). ■

From the Log of the Mark V Home Computer & Intruder Alarum

```
DANGER! DANGER! DANGER!

I, THE MARK V HOME COMPUTER/INTRUDER ALARUM DETECT A SUSPICIOUS
LOOKING INDIVIDUAL ATTEMPTING TO FORCE ENTRY TO THE BUILDING.
COMMENCING INTRUSION LOG:
    18:33:47.023 FRONT DOOR INTERLOCK BREACHED
    18:34:54.543 PHOTOCELL CHECKPOINT ALPHA PASSED
    18:36:06.105 WEIGHT OF 97 KILOGRAMS DETECTED ON FIRST STAIRSTEP
    18:37:22.133 INTRUDER ON LANDING; DIGITAL PHOTOGRAPH TAKEN
    18:37:22.354 PHOTOGRAPH STORED AND PROCESSED. ANALYSIS INDICATES
                 WHITE ANGLO SAXON CATHOLIC WITH SCAR ABOVE LEFT EYE
                 AND A SLIGHT LISP.
    18:37:22.665 PHOTOGRAPH DISPATCHED VIA TELSAT TO INTERPOL.
    18:37:22.982 INTERPOL RETURNS ID AS "GREGOR TABRASKII", KNOWN
                 COMPUTER THIEF AND KILOBAD SUBSCRIBER.
    18:37:22.995 ENERGIZE DEFENSE SYSTEM.
    18:37:23.442 TARGET ON FIFTH STEP OF SECOND TIER.
    18:37:23.445 DEFENSE SYSTEM READY.
    18:37:25.045 TARGET AT TOP OF STAIRS. LOCK ON 50 MEGAWATT LASER.
    18:37:25.050 LASER LOCKED ON. QUERY SUPREME COURT ROBO-JUDGE FOR
                 PERMISSION TO TERMINATE LIFE OF INTRUDER.
    18:37:25.052 PERMISSION GRANTED.
    18:37:25.053 ZAP!
    18:27:25.054 SWITCH AIR RECIRCULATION SYSTEM TO HIGH CLEAN.
    18:37:25.100 NOTIFY BUILDING ENGINEERING OFFICE OF NEED TO
                 REPAIR FRONT DOOR AND REAR WALL OVER STAIRS.
    18:37:25:125 RESET ALARUM SYSTEM, RETURN LASER TO STAND-BY.
    18:37:25:143 TERMINATE INTRUSION LOG.

MARK V, READY
```

SOFTWARE TECHNOLOGY MUSIC SYSTEM

David H. Ahl

The opening page of the Software Technology Music System says, "Perhaps you have a computer and have been wondering: 'What else can I do with it?'

Perhaps you are a musician and would like to investigate computer-generated music.

Perhaps you would like to play a concert."

Well, I am none of these. I have plenty to do with my computer, I am certainly not a musician and, although I would like to play a concert, I hardly have the time to devote to doing so. Some of my friends have occasionally accused me of being a baroque freak because when they come over to the house all they can seem to get on my hi-fi is Bach, Handel, Vivaldi and so on. However, my musical tastes actually run from Baroque to the Beach Boys which encompasses a fair amount of ground in-between. As a result I was intrigued with the flyer about the Software Technology Music System for $24.50.

The price covers a very small piece of hardware (a printed-circuit board about 1½"x6" which plugs into a standard S-100 bus and which mounts all of five different parts, three resistors and two capacitors), an instruction manual, and a cassette tape of the software. It sounded like an intriguing investment for only $24.50.

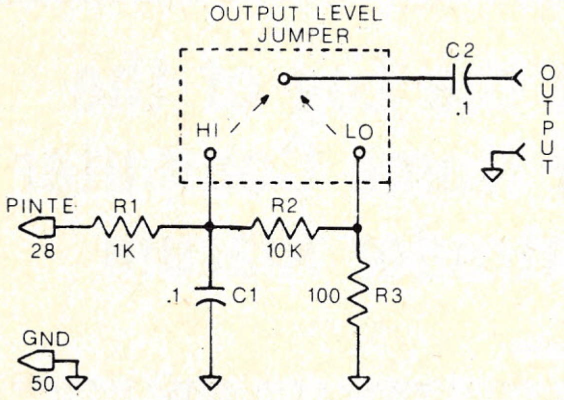


Fig. 1. The hardware is simplicity itself; the entire board has only five components.

My last investment in a music kit was also, curiously enough, in the same price range ($30). That was for a dulcimer which I purchased several years ago. The dulcimer might be termed all-hardware; perhaps hardwood would be a more accurate designation. In addition to the price I found several other interesting parallels between the dulcimer and the Software Technology Music System. The dulcimer has three strings — the music system has three voices. It took me, a totally inexperienced musician, a couple of hours before I could start picking out simple melodies on the dulcimer and making them sound decent using all three strings. Correspondingly, it also took a couple of hours before I felt relatively comfortable transposing music for the Software Technology Music System. The dulcimer has rather simple tuning, a two-octave range, an interesting tone if played in moderation, and is kind of a folksy musical instrument. I found the Software Technology Music System roughly the same. It's easy to tune, has a wider range (four octaves) with a nice tone although one certainly wouldn't want to overdo it, and it's kind of folksy — something that my wife and kids, who don't relate to computers all that well, found intriguing and could relate to. Enough of the parallels. What is the product like?

The printed-circuit board clearly is the model of simplicity. It goes together in about ten minutes — five minutes of which are spent letting the soldering pencil warm up. The addition of a shielded cable and a phono plug allows you to plug it into your hi-fi. In my case, since I was using a tv set for my display with my SOL-20 system, I decided it would be easy enough to tap into the volume control on the tv set and use its amplifier and speaker for the music system. Certainly not the fidelity of a good hi-fi system but adequate under the circumstances, plus the fact that it gives me a fully-transportable, self-contained system. In the way of other hardware, you need just the bare minimum system: an 8080-based CPU, with preferably 12K or 16K of memory since a typical musical tune takes approximately 1K of memory for each minute of the composition. If you don't want to overlay the compiled version of the tune on top of the source code you would certainly want at least 12K of memory for any tune more than a couple of minutes long.

The cassette tape of software comes in two different formats: a 1200-baud CUTS format on one side and on the other side the same programs in the 300-baud Kansas City format. The tape contains seven programs. First of all, the music program itself in object form, and six musical scores in source form including works such as the *Prelude in C Major* by J.S. Bach, *Bouree* by Handel and several other baroque pieces.

Upon assembling my system the first thing I did was check it out and make sure the tape read in correctly; in other words, I turned to page 24 in the manual, entered a few commands to my SOL-20 system to load and execute the music program, return to the SOLOS monitor and get the first demo program, then gave it the command "FILE" which verifies the new file just read in, then the command "SCORE" which compiles the file, and finally the command "PLAY" which plays it through the speaker system. Frankly, I was absolutely flabbergasted. The system loaded absolutely correctly the first time and played through my hi-fi system the first time. I had visions of spending the usual one or two evenings debugging the system, figuring out why it didn't load, why it didn't play, and so on, and frankly I was pleasantly surprised that the whole system loaded and played the first time.

Flushed with this extraordinary success I decided to try my hand at putting in a piece of music of my own. As I mentioned before, I am anything but a musician, nor are any of the people in my family musicians. Consequently the only music we had around were some things that were from the 1940's left over from when other people dropped over occasionally to play our piano and forgot to take their music home with them. In other words, our selection was rather limited. In this pile I found a piece of sheet music for

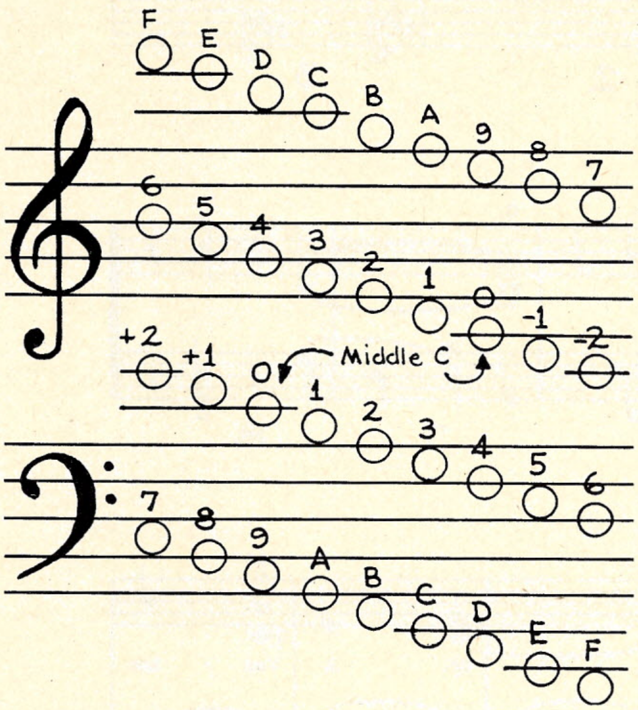


Fig. 2. Musical note tones are expressed in hexadecimal with notes on the treble clef being positive and on the bass clef negative.

Yankee Doodle Dandy by George M. Cohan in "an easy arrangement for piano." It suited my purposes perfectly because the arrangement was for three parts and the music board itself allows for three parts or three voices. I arranged things on my desk, the musical score in the center, a writing tablet and the manual for the music system on the left and started away. I first started by numbering the measures of the piece as was suggested in the music manual. As it turned out there were 42 measures in *Yankee Doodle Dandy.* Then I simply followed the directions and transcribed the music — the various notes, key signatures and so on from the sheet music to the appropriate computer notation. After completing eight measures I typed in the code and scored it, which is simply the command "SCORE" in the music system and played it. I expected to recognize the melody. Instead I found that I was playing some sort of funeral dirge. "Aha!" I said to myself. The speed must be wrong and so I tried to modify the speed by means of the command that fixes a certain number of computer cycles to a certain note length. However, affixing the minimum number of cycles to the longest note, in other words playing the piece as fast as I possibly could, while it didn't quite have the funeral quality when I started out, was still far too slow for comfort. My solution was to combine two measures into one and half the length of each of the notes. In other words, on the sheet music a quarter note became on my computer music an eighth note, a whole note became a half note and so on. This gave me the latitude that I needed and improved the tempo of the piece considerably. While *Yankee Doodle Dandy* is a rather simple piece, particularly compared to *Sarabande* which is the sample piece scored in the Music System manual, nevertheless, *Yankee Doodle Dandy* had several musical occurrances which were not in *Sarabande.* Consequently I found myself looking in the manual to find out how to do certain things. Clearly in a system such as this, once the hardware is functioning and the software is loaded in, it is the manual which becomes the central part of the system. Consequently a few comments on the manual are probably in order.

It would certainly be helpful to have all of the symbols of the music system language defined in one place. As it is right now some of them are defined in the "commands" section of the manual, some others are defined in the "language summary" section, while still others are defined in the "musical note symbols" section, and lastly there are some symbols which are defined only in the text. The "rest" symbol, which is a dollar sign, turns out to be very important in *Yankee Doodle Dandy.* It's a rather minor part of *Sarabande* and is treated lightly in the text. On the other hand it would certainly have helped to be able to refer to some part of the manual and find out easily how to use the rest. Another thing which is not discussed in the manual is how to tie a note over from one measure to the next. The manual states that "since the computer always plays 'legato,' tied notes usually do not present much of a problem. However, a very soft, almost inaudible 'sh' sound is produced by the tone-generating routines whenever a new note is started." The discussion refers to two notes in the same measure. The question the user must ask himself, as I did, is "does that mean that when the same note is played one measure after another that it is also played 'legato' or does it mean that there is a bit of a pause or articulation between the two notes?" As it turns out, trial and error indicated that things are always played legato and if one wants a break between the same note played in two measures a very distinct articulation must be inserted at the end of the note played in the first measure.

After several hours wrestling with these various problems the system became clearer and clearer; trial and error was a most helpful friend for determining what would happen in different situations; and, as the manual says, "it will be a matter of blind luck or painstaking trial and error. Ultimately, the 'right' solution is the one that sounds best." The manual is absolutely correct in this regard. After several hours of trial and error I found that instead of transposing the music to a piece of paper and then into the computer that I could simply sit down at the keyboard and transpose the music directly from the musical score into the computer.

Later that day some friends were over to the house. This happened to be the Fourth of July and we had a few firecrackers to drink, not the exploding type (the drinkable type of firecracker is made from a jigger of whiskey in a tall glass with ice filled up with cranberry juice — a delightful summertime drink). In any event I turned on my music system, played some of the demonstration works and then said, "here is the first work that I composed myself" and, as those old ads say, they laughed when I sat down at my computer (and are probably still laughing yet!)

All in all, if you're willing to devote a couple of hours to putting together the hardware, playing the demo software, and learning how to transcribe some music with trial and error or whatever method you find most handy, you will find "The Music System" by Software Technology an extremely intriguing and worthwhile investment of $24.50.

For more information on the Software Technology Music System, write: Software Technology Corp., P.O. Box 5260, San Mateo, CA 94402. Phone 415—349-8080.

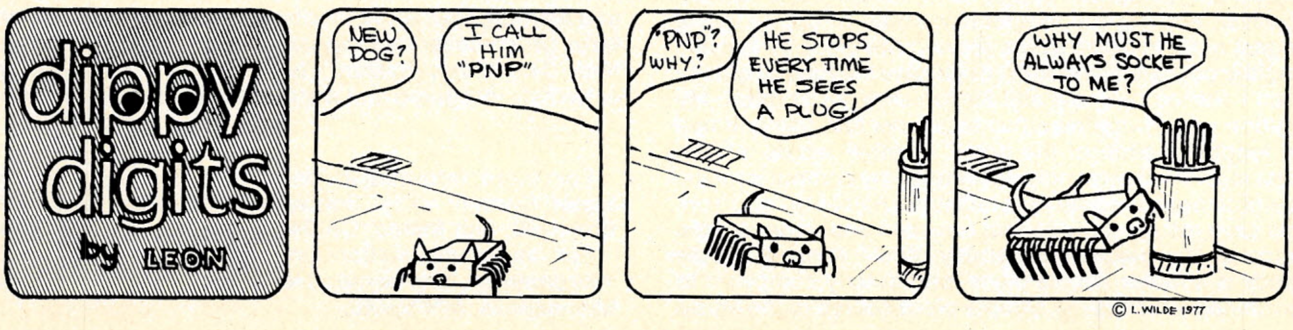
For more information on dulcimer (and other folk instruments) kits, write: Here, Inc., 410 Cedar Ave., Minneapolis, MN 55440.

YANKEE DOODLE DANDY

(The Yankee Doodle Boy)

GEORGE M. COHAN

Fig. 3. First page of the piano score for *Yankee Doodle Dandy*.



Music Transcription Example

A portion of the piano score of *Yankee Doodle Dandy* is shown in Figure 3. The computer source code is in Figure 4. Let's consider various lines of the computer code.

0020. A slash indicates a comment, and the rest of the line is ignored.

0040. 4 means that all the notes are transposed down four semitones. Thinking of the computer as a musical instrument the trick, as with any instrument, is to find the key which sounds best. In this case, four semitones down sounded 'right.'

0050. Defines the key signature, in this case one sharp.

0060. Defines the tempo. In this case, a sixteenth note (S) has 40 computer cycles. NS=90 would play the tune more slowly.

0080. Indicates that the measures that follow are one section of the piece (Part A). Part A will consist of everything until another P symbol. Note that Part D (line 310) is a repeat of Part B (RB).

0090. M1 defines the measure. It is ignored by the computer but is handy for debugging. A voice (or part) is a separate strand of music, in harmony or counterpoint, as in a three-voice fugue. Up to three voices may be defined in a measure, identified as V1 (assumed to be the first voice in a measure), V2, and V3. V1 usually carries the melody, or highest notes, V3 carries the bass, and V2 the mid-range voice.

Each note is represented by a pair of symbols, one of which defines the tone (position on the staff) and the other the length (whole, half, quarter, etc.). As mentioned in the main text, the note lengths were halved to give the piece adequate tempo, hence a quarter note in the music becomes an eighth note in the computer transcription. Also, one computer-music measure corresponds to two measures in the score. I4 means an eighth note of tone 4 (i.e., a treble G). The leading asterisk (*) indicates the treble clef. The comma following indicates that the note is to be played staccato. Voice 2 is in the bass clef ( ) and consists of two half notes of tone 6 (bass D). Voice 3 is also in the bass clef and consists of two half notes of tone A (bass low G).

Dotted notes are indicated by a period (.) after the letter indicating primary note length (see lines 0110 or 0140). A colon (:) indicates triplets. A dollar sign ($) indicates a rest (see lines 0120 or 0170).

Accidentals are indicated with a sign following the note to be modified, for example HO# (C sharp) and HO% (C natural) in line 0110.

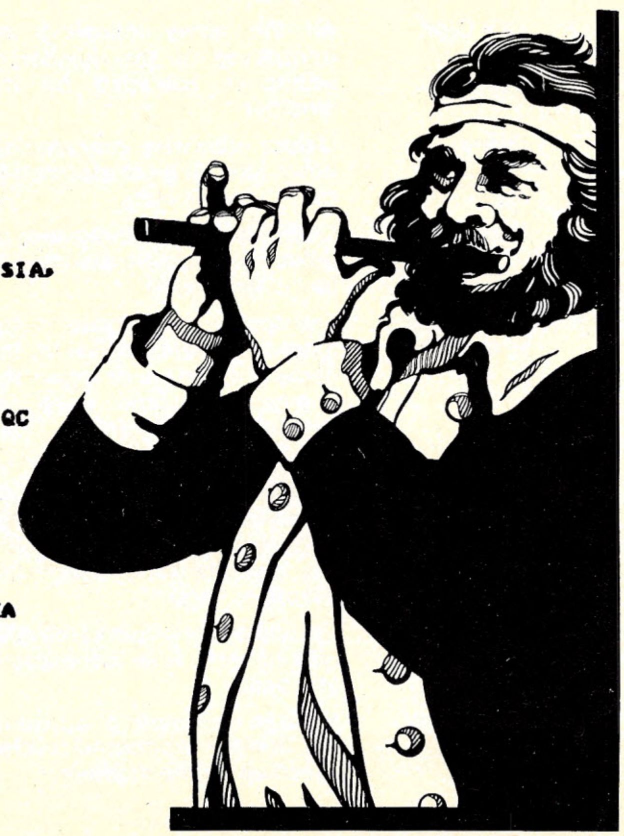
0120. The length of a measure is equal to the longest voice in it. Shorter or undefined voices are filled with rests. Hence it is up to the user to keep track of beats and indicate rests for at least one voice to keep the proper tempo. Hence Voice 1 closes with I$ (eighth rest) in measure 8.

0140. Between measures 9 and 10 it is desirable to have an articulation (between "a" and "Yan" which are both A) otherwise they will sound almost as one long note. A double quote (") gives a long articulation equal to about two-thirds of a sixty-fourth note. The duration of the preceding note is reduced to compensate for the added rest.

Many other things that occur in various pieces of music can be handled in the Software Technology Music System. Even in this piece a better job could have been done with the short bass c-sharp in the opening four measures (instead of ignoring it). However, as with any instrument, proficiency comes with hours of practice.

```
0020 /YANKEE DOODLE DANDY BY GEORGE COHAN
0030 /COMPUTER TRANSCRIPTION BY DAVID AHL
0040 <4
0050 K1#
0060 NS=40
0080 PA
0090 M1 *I4,I4,I5,I6,I4,I6,I5,I1, V2@H6H6 V3@HAHA
0100 M3 *I4,I4,I5,I6,Q4Q1 V2@Q6Q6H6 V3@QAQAHA
0110 M5 *I2Q5I2I.3S4I5,I3, V2*HO#HO% V3@Q3Q5Q6QD
0120 M7 *H4I4I$IB,I$ V2*H-1I-1I$I6, V3@IAQ5I6I3,I$IA,
0130 PB
0140 M9 *Q.6I5"I5I4I3I4 V2*H1H1 V3@IAI6IDI6H8
0150 M11 *H5Q.2"I2 V2*HO#HO# V3@I9I3I5I3I9I3I5I3
0160 M13 *Q.5I6I5I3I2I1 V2*HOHO V3@I6I7Q8Q6Q4
0170 M15 *H4Q.4"I6 V2*I$I1I$I1I$I1 V3@Q3Q5Q6I8
0180 M17 *Q6"Q6I4#I5I6I8 V2*H1Q1Q4 V3@IA#I5ICI5Q8QC
0190 M19 *Q7Q6H5 V2*Q2Q1HO V3*I9I5IA#I5I9Q5"I5,
0200 M21 *I6Q5I4I2Q4I6 V2*WO# V3@Q3Q5Q3Q5
0210 M23 *H5Q.5I1 V2*HOQ.0 V3@I4Q6I5I4I5I4
0220 M25 *Q.6I5"I5I4I3I4 V2*H1H1 V3@IAI6IDI6H8
0230 M27 *H5H2 V2*WO# V3@I9I3I5I3I9I3I5I3
0240 M29 *Q.5I6I5I3I2I1 V2*HOHO V3@I6I5Q4Q6Q4
0250 M31 *W4 V2*I$I1I$I1I$I1I$I1 V3@Q3Q5Q6Q8
0260 M33 *I4,I4,I5,I6,I4,I6,I5,I1, V2@H6H6 V3@HAHA
0270 M35 *I4,I4,I5,I6,Q4Q1 V2@Q6Q6H6 V3@QAQAHA
0280 M37 *I2Q5I2I.3S4I5I3 V2*HO#HO% V3@Q3Q5Q6QD
0290 PC
0300 M39 *H4I4"I4I5I5# V2@H1I1 V3@IAI6I5I4I3
0310 PD RB
0320 PE
0330 M41 *H4I4I$IB V2*H-1I-1I$I6 V3@IAQ5I6I3I$IA
```

Fig. 4. Musical source code for *Yankee Doodle Dandy*.



Music Language Statements

Symbol	Modifier	Meaning
/	(None)	All characters on the rest of line are ignored.
P	Any letter (A-Z)	Define beginning of a part identified by the modifier. Any previous part is ended.
R	Any letter (A-Z)	Repeat the part named by the modifier.
M	Any character or characters	Define the beginning of a measure. Any previous measure is ended.
V	Digit (1,2, or 3)	All the notes following belong to the voice named by the modifier.
<	Hex Digit (0-F)	All the notes following are transposed down the number of semitones specified by modifier.
>	Hex Digit (0-F)	All the notes following are transposed up the number of semitones specified by the modifier.
*	(None)	Unless otherwise indicated, all notes following are assumed to be "+" (treble clef).
@	(None)	Unless otherwise indicated, all notes following are assumed to be "-" (bass clef).
↑	Signed Hex (+ or -) (0-F)	Transpose only those notes following that belong to the current voice up or down the number of whole steps indicated in the modifier.
K	Digit Char. (0-7) (# or &)	Key signature is defined by number and type (sharp or flat) specified in the modifier. If this symbol group is omitted, the key defaults to C major (no sharps or flats).
N	Char (H,Q,I,S)	Correlates the length of the note type in the modifier to the length of a beat.
=	2 Digit Hex (00-FF)	Equates the length of a beat to the number of internal cycles specified by the modifier.

Music System Commands

Commands can all be abbreviated to only the first letter. Portions in parentheses are optional operands.

NEW (addr1)	Establishes a new file beginning at 'addr1'.
FILE (addr1)	Validates the file beginning at 'addr1'.
LIST (line 1 (line 2))	Displays the lines in the current file between 'line 1' and 'line 2'.
DELETE line 1 (line 2)	Deletes all lines between 'line 1' and 'line 2'.
SCORE (addr1)	The current file is compiled, and the resulting binary object code is entered into memory beginning at 'addr1'.
PLAY (addr1)	Generates musical tones according to the binary code at 'addr1',
RESERVE addr1	Marks the end of the memory space to be used by the music system.
RETURN	Returns to SOLOS (or to a properly written surrogate monitor).

Note Specifications

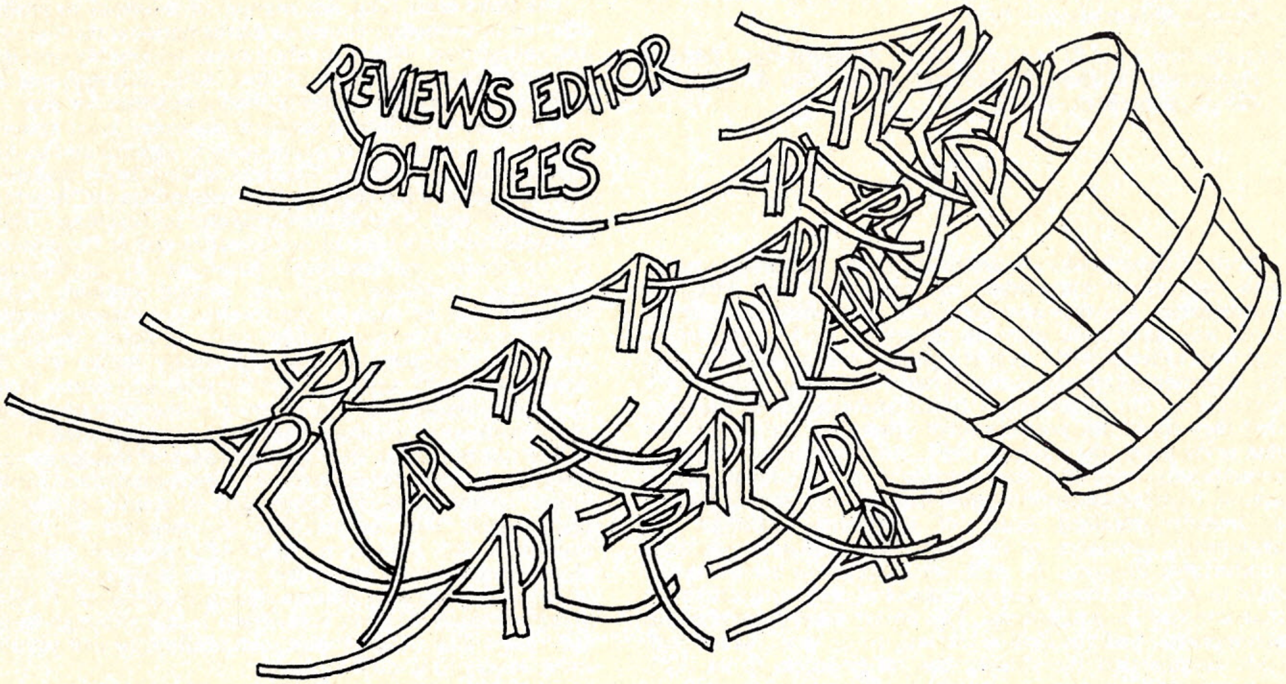
Note Modifier	Name	Musical Example
#	Accidental Sharp	#
&	Accidental Flat	♭
%	Accidental Natural	♮
'	Short articulation	none
"	Long articulation	none
,	Staccato	

Note Value Symbol	Name of Note	Musical Equivalent
W	Whole note	𝅝
H	Half note	𝅗𝅥
Q	Quarter note	♩
I	Eighth note	♪
S	Sixteenth note	𝅘𝅥𝅯
T	Thirty-second note	𝅘𝅥𝅰
X	Sixty-fourth note	𝅘𝅥𝅱

Note Value	Name	Musical Example	Time Value Multiplier
.	Dotted note	♩.	1-1/2
:	Triplet		2/3
$	Rest		

■

A Bushel of APL



The Reviews Editor gave me a bundle of publications on APL all at once, so it seems proper to do a composite review. But first it might be appropriate to introduce APL to those who are not familiar with it. So far it has not swept the microcomputer field, but the entry has begun, and APL has generated as much heat and argument as any computer language. It has raised antagonism and won fanatic devotees and we should at least be aware of APL's existence.

APL began with Kenneth E. Iverson's book *A Programming Language* (Wiley, 1962). Hence the name APL. So many of the hot topics of today — microprogramming, stack computers, associative memories — are covered in that book that one despairs how slow, not how fast, progress in computing is. Iverson then began working with Adin Falkoff at IBM to implement the language. In that process the syntax of APL was modified to its current form to make it more suitable for computer use and implementation. In the meantime APL was used in publishing the formal definition and description of the then-new IBM System/360 series of computers in 1964. In 1966 the first implementation, *APL/360*, became available to users.

By design, APL is a language to be used from an interactive terminal. Editing, tracing, error messages, and other debugging facilities are an integral part of the language, not *ad hoc* facilities added more or less well, and differently, to each implementation. It is also a language inherently suited to run-time interpretation, rather than compilation and subsequent execution of compiled code. Indeed, it is generally claimed that APL cannot be compiled at all. However, it can benefit greatly from the inclusion of special facilities in the microprograms of host computers as, for instance, is the case with the *APL ASSIST* feature on later models of the IBM System/370. I predict that soon you will be able to buy off-the-shelf ROMs for most microcomputers preprogrammed with APL interpreters. But you will also need up to 64K bytes of working storage and a terminal with the APL symbol set. This latter point has some people upset, APL uses its own symbol set which has over 60 special characters in addition to the alphabet and digits. Some of the special characters are overstrikes, composites of two symbols. While implementations of APL that allow the use of ordinary ASCII terminals exist, I cannot imagine using it this way. In ASCII the special APL symbols have to be represented by the dollar sign trailed by two more or less mnemonic characters, and the whole flavor of the language, and the concise and clear statement form, are lost.

The "natural" data item of APL is the array, of any rank. Scalars, of rank 0, are a special case and sometimes behave rather differently from what would seem to be the same, a vector of one element only. The basic scalar data types of which arrays may be composed are numbers (some implementations distinguish between integers and floating-point numbers internally, but transfer functions are applied automatically), characters, and logical values. The scalar functions of APL are so called not because they only take scalars as arguments but because they operate on arrays element by element, while the mixed functions may operate on arrays as a whole. All APL functions defined by the user are potentially recursive. They may have local variables and the scope rules for these are dynamic, not the static scope rules of block-structured languages. If you think about it, this is a necessity with the interpretive nature of APL. The "unit" of APL is not a program, as in other languages, but the workspace. This may contain both variables and user-defined functions. A workspace may be filed and retrieved without harm to the values of its variables; indeed, it is possible to suspend function execution, save the workspace, and later retrieve it and resume execution.

Currently, APL is available on most timesharing systems and one worldwide service. I. P. Sharp & Associates Ltd. and Scientific Timesharing Corporation offer no other language on their bureau machines. Among IBM (US) employees there are reputed to be 28,000 active users of the language. It is taught at many schools and colleges. I know of one program in which all students at a nearby junior high school learned APL for some years, until the school board cut the funds for computer access!

Now to the books.

★

APL An Interactive Approach. Leonard Gilman and Allen J. Rose. John Wiley & Sons Inc. 378 pp, paperback. $11.95. Second edition revised, 1976.

Introduction to APL and Computer Programming. Edward Harms and Michael P. Zabinski. John Wiley & Sons Inc. 400 pp, paperback. $10.95. 1977.

☆

Both of these books are addressed to the beginning student with no prior knowledge of computers or APL, or mathematics. But they also serve well as teaching manuals for those who have some previous experience with programming and wish to learn APL. At first sight this may seem an insult to the experienced programmer. These, and most other, APL texts are so full of seemingly trivial and repetitious exercises that they could not possibly serve other than the novice. But that is the point, the novice at APL. All of us carry a burden of assumptions from our first introduction to mathematics in school. Expressions are evaluated from left to right and there is a divinely ordained order of precedence amongst the arithmentic operators, or functions. APL violates both these comfortable illusions. In APL all expressions are evaluated right to left, and there is no precedence of operators other than that imposed by parentheses. The novice must shed previous assumptions, and the drill is the way to do that. Hence both texts assume that the student has ready access to an APL terminal and system to complete the exercises and to experiment.

It is difficult to choose between the books; any choice is probably a matter of personal taste. We do have some years' experience with Gilman and Rose with our own students, and they have no complaints about it. The other book is too new to assess that way.

While Gilman and Rose tend to introduce examples of APL applications as soon as the necessary functions and operators have been explained, the Harms and Zabinski book is in two parts, the first defining the language and the second giving examples of its application. The examples are also much better indexed. Both books cover examples ranging from mathematics thru scientific and engineering applications to data processing and commerce. Users of implementations on other than IBM systems will need their particular manuals in addition to either book to cover local log-on procedures, mass-storage facilities, and some tracing and debugging operations that relate to the system, and for other local system features.

The student exercises in both books are very comprehensive but where Gilman and Rose offer solutions to all problems, Harms and Zabinski provide only solutions to every second exercise.

★

The rest of the items to be reviewed are all from APL Press, Box 378, Pleasantville, NY, 10570. Payment must accompany any order, except for schools and libraries, but unlike most publishers APL Press offers quantity-purchase discounts to private individuals, on the following scale of total single order value: over $30 - 15%: over $100 - 20%; over $300 - 25%; over $1000 - 30%.

ALGEBRA: An Algorithmic Treatment. Kenneth E. Iverson. Addison-Wesley Publishing Co. and APL Press. 361 pp, paperback. $9.35. 1972.

Solutions to Iverson's Algebra. Janet A. Iverson. APL Press. 42 pp, paperback. $1.50. 1976.

Elementary Analysis. Kenneth E. Iverson. APL Press. 218 pp, paperback. $6.25. 1976.

CALCULUS in a new key. D. L. Orth. APL Press. 286 pp, paperback. $8.00. 1976.

☆

The three texts (and one manual of solutions to problems) offer an introductory course in mathematics at the high-school and college level. Alternatively, anyone reasonably confident in mathematics can use them to learn APL, although one of the texts reviewed earlier would probably be better for that.

The three texts form a series in the order listed above. The student is introduced simultaneously to algebra and to APL in the first book by Iverson. Teachers, and probably students too, should first read the two excellent appendices on "Algebra as a Language" and "Use of the Computer in Teaching." The first explains Iverson's view of mathematics and of the importance of particular mathematical notations in explaining, or obscuring, problems, and the second outlines how the computer can be used by the teacher to enhance comprehension of material in many subject areas. If, that is, the computer can be programmed in a pedagogically useful way. While access to an APL system is not essential, any course based on these texts should really be planned on the basis of regular use of the computer to complete the problems presented.

The mathematical style and approach of this series may be offensive to an older generation of mathematicians for whom vulgar computation, the generation of results, is best left to clerks while they contemplate the elegance of their formulations. But it turns out that APL notation does lead to elegant mathematics and is very helpful in suggesting solutions at the same time as it ensures rigorous expression. It also aids comprehension. My son is in Grade 7, and although he would still rather play football than read algebra, he can yet already make sense of the early chapters.

The other two volumes continue on from the introduction to algebra; here there is of course less emphasis on APL as a language and more on the mathematical treatment. Whether the approach chosen here is better than the conventional one cannot be judged out of thin air, it would require the opinion of teachers who have used the older approach and then these texts, after becoming sufficiently familiar and skilled in APL.

★

APL in Exposition. Kenneth E. Iverson. APL Press. 61 pp, paperback. $1.00. 1976

Introducing APL TO Teachers. Kenneth E. Iverson. APL Press. 25 pp, paperback. 75¢. 1976.

An Introduction to APL for Scientists and Engineers. Kenneth E. Iverson. APL Press. 26 pp, paperback. 75¢. 1976.

☆

These three booklets were originally published as IBM Technical Reports in 1972 and 1973. They are aimed at introducing teachers and members of the scientific and engineering community to APL.

APL in Exposition gives a basic introduction to APL and then devotes 6 to 10 pages each to illustrations of the use of APL in teaching various topics ranging from coordinate geometry

thru logic to electric circuits. For computer buffs the most interesting things are a complete simulator for an 8-bit-wordsize, 8-instruction, 32-word-memory computer in two APL functions comprising 14 lines of code. An assembler for a symbolic assembly language for this machine to translate mnemnic opcodes and symbolic addresses to binary is given in seven functions totalling 19 lines. Or, if you would rather have a simulator that executes simple APL functions directly, that takes a function of a mere nine lines, but needs 19 auxiliary functions of 30 lines in all to transform APL statements into its required internal representation. By now you might have guessed that APL is indeed a very concise language! User-defined functions of but a single line, doing useful work, are common, even tho APL users have outgrown their early childish delight in the "one liner," the lust to write as complex and powerful a statement as the language will allow, without regard to comprehensibility.

Introducing APL. . . is largely a set of exercises, intended to be completed at the terminal in 15 to 20-minute sessions, five to six hours in all, for teachers who wish in turn to introduce their students to the computer and APL. The exercises come in sets, each on a particular topic, designed as a terminal session per set. The first one or two exercises of a set come with solutions to reinforce the student's confidence, and the next one or two draw out the user's skill in applying the knowledge already gained. The topics covered are those of interest in teaching mathematics and science in high school, graphs and power functions and statistical measures and the like.

An Introduction to APL. . . builds on a knowledge of vector algebra to teach APL and the use of the computer in scientific and engineering applications. Somehow this booklet lacks the sparkle and enthusiasm of the others. It is a potboiler, with examples too pedestrian and narrow in scope. Or mayhap that is a reflection of the author's estimation of that particular audience! But it still conveys the message that a large part of learning APL, and the potentialities of the computer, is to experiment, to play, at the terminal.

★

APL Reference Card. APL Press. 25¢. 1976.

☆

This vestpocket-size plastic reference card lists and gives examples of use of all the APL scalar and mixed functions, operators, system variables, and system functions. To say that it is a most useful tool to refresh the memory when sitting at a terminal, or jotting down a function, is to miss the point completely. Here we have what must be one of the most powerful and complex programming languages in existence, yet its essential points can be compressed onto the two sides of a 9.5-cm by 6-cm plastic card. So much for the vaunted orthogonality of ALGOL 68.

★

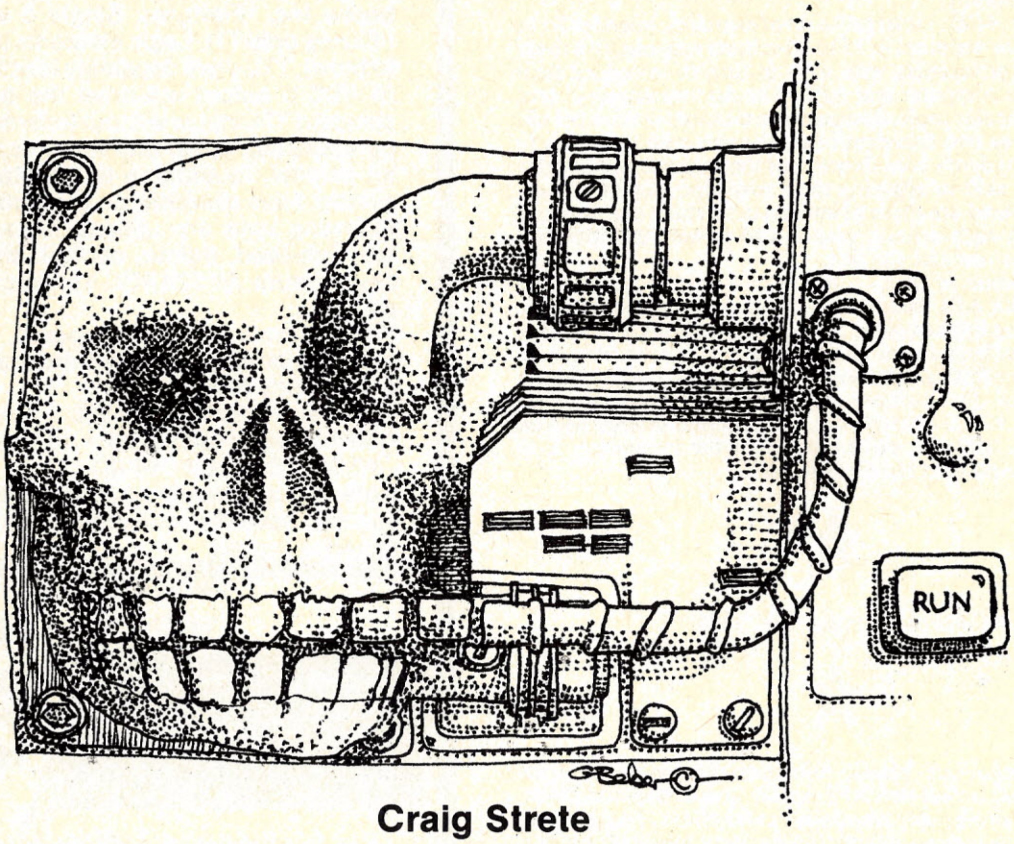
APL News. APL Press. 8 pp per issue.

☆

This is a periodic newsletter available on demand from APL Press. It publicizes their publications, offers space for interested users to explain novel applications or suggest improvements to the language, publishes APL games and puzzles, explains interesting techniques, gives notices of relevant meetings, etc., etc.

H.D. Baecker
Calgary, Canada ■

Into Every Rain A Little Life Must Fall



Craig Strete

I punch into the console web, link in to the main computer. The control room is warm and comfortable but outside it's a miserable night. The street monitors sweep my sector and all of them shoot back the same story. No action.

I'd lucked out on assignment. Hit the graveyard shift which is my favorite. Most of the action breaks at night. Not tonight though.

It's cold and it's raining to beat hell and this is one of those kinds of nights that give me the womb cop blues.

The streets in my sector are deserted. Very depressing. I like action. I sit there behind my monitors, audio helmet jammed on my head, feel like a football player sitting out a game on the bench.

I dialed Central to report myself in. "WOMBCOP 345-45. STEVENS, ROGER DAVIS. Reporting for duty, shift 2, punch in 0200, all systems functioning, nothing to report, no shift 1 carry-overs."

It was a slow night all over. I had only about half of my mobile street units out. Rain had the whole city locked in. It was coming down hard and cold and nobody in his right mind was out in it, or anyone in his wrong mind either.

My hands itched with inaction, toying with the trigger grips of my bank of pocket lasers.

The rain had cut down visibility and I had all dispatched scanners turned up to the highest wide angle scoop. Even then, my visual range was pretty limited.

I don't feel useful on a night like this. I like the action, like the feel of being on top of a crime, hitting into it, punching it in and putting it down. Then if I'm lucky, burn down. I wish there was some way of expressing the satisfaction I get when I burn down a criminal. I love my work.

Fifteen minutes plugged into the computer and not one peep.

Then action. "Position," said the computer. "Pickup 27, Monitor 7."

This is more like it! I punch in video and audio and man I feel alive again!

Nothing on audio but the sound of rain coming down on the pavement so hard its bouncing. I tap the toggle on my helmet. I'm turning up to high gain. Still nothing but the damn rain.

Visuals, the same story. A grey side street shrouded in

rain. Can't pierce the rain more than ten feet at a time. I link into the mobile unit. Scanners on high scope, still can't see a damn thing.

"27-7, move toward subject!" The monitor begins moving down the street, rapidly.

The computer reads out, "Pedestrian, Unidentified Racial Type, Unidentified Gender. Computing."

"Identify," I snarl. I can't even begin to guess what's coming down.

The computer hesitates and then again. "Pedestrian, Unidentified Racial Type, Unidentified Gender. Computing."

"Move in close, damn it!"

"Acknowledged."

I tap the trigger grips impatiently. This seems like it's taking forever. I feel like I am playing pin the tail on an invisible donkey.

Finally, audio picks up the sound of footsteps, the sound of feet splashing through puddles. A fraction of a second later, video picks out a bedraggled figure moving slowly through the rain. Heat scanners must have sensed him a long way off.

"Identify." The scanners freeze frame his face, code and transmit the image automatically to Central.

"Caucasian, Male. No information. No identity card, no arrest record. It does not compute."

Has to be a computer foul up. Maybe fifty years ago it might be possible for someone to exist without an identity card but not now. Somebody in programming deserved a long vacation without pay.

"Pursue and monitor." I order, stalling until Central rings in with the correct information. That's the best I can do.

"It does not compute. Lack of data." Clacks out Central.

"Telephoto zoom. Target, hands and fingers. Positive print I.D. check." I order the mobile unit which immediately begins circling subject, clicking extreme close up telephoto freeze frames. I punch in the information direct to Central.

"Information acknowledged." reads out Central. "No print record. Information does not compute."

What can I do? Damn programmers! I punch in. "Check programmer error!"

Central beat me to it. "Possibility programmer error eliminated. No identity card. No file tapes. Detail and Identify. Violation of Identity Code, Section 348. Hold for questioning." One entire panel lights up on my console. My computer units all lock into Central. They're functioning full gage on this one. Damn!

I've been a womb cop for ten years, ten years and I've never run into anyone who didn't have an identity card, who didn't have an identity tape on file! It's not only illegal, it's damn impossible! This was something new we had on our hands.

Two more panels switch in. The computer is going crazy on this one. As far as it's concerned, the impossible has happened.

I've got my eyes riveted on my monitors and I'm really giving our boy a looking over. He's no beauty.

"Detain." I punch in and the mobile unit who has been keeping pace with this character moves in and cuffs him to the detention cable on the side of the unit. No resistance, no reaction at all. Subject seemed unaware of the monitor circling around him.

It's an old man, video observation indicates. Frayed overcoat. About 5'4", pants too big and ragged. Looks like an alcohol addiction case, a wino, unshaven. Eyes, on full zoom, look bloodshot. He's unconcerned. Looks like he doesn't care one way or the other about being stopped. Alcohol probable cause of brain damage indicated by subject's lack of interest, negative display

"Possibility programmer error eliminated. No identity card. No file tapes. Detain and Identify. Violation of Identity Code, Section 348. Hold for questioning."

of emotional response.

"Who are you? Please identify yourself?" My voice comes through the mobile unit speakers. Tapes being filed, a direct line to Central. All my panels are lighting up. My console looks like a computer light show. Central is really shooting sparks over this.

The old scarecrow looks directly into the monitor. Gaunt features, eyes sunk into his head. Deathly white face. I'd swear I was talking to a corpse. No expression on the face, just kind of cold and withdrawn. No answer.

"Repeat. This is wombcop Davis. You are in violation of the Identity Code, Section 348. Please identify yourself."

Not a flicker of anything from the old man.

Central punches in. "Section Commander Hartmann on the line. What the blue hell is going on down there?"

I beep in acknowledgement of his call.

"Checking, sir. We have a man with no identity records, sir."

"That's impossible!" Hartmann sounds fit to be deprogrammed.

"Please identify yourself?" I try again. Jesus, this is really one for the tapes!

"Plug in your lie detector monitors!" snapped Hartmann, his voice booming through loud on the line.

"They're already plugged in, sir! I can't get a response from subject, sir." Damn, I feel like an idiot. He knows I haven't got a response, that order about the lie detector was just to prod me into getting one. This action is plugged into every section of Central. My console panels flash with a thousand simultaneous plug ins. Everybody's interested in this one.

My eyes stayed on the monitor. The old man turned away from the monitor and looked back over his shoulder, as if looking for someone, as if someone were following him.

"It's raining," said the old man. He turned around and looked straight into the monitor again.

I went to split screen, turned the console camera on me and put my picture in the bottom half of his screen. Standard interrogation procedure.

"This is wombcop Davis. You are in Violation of . . ."

He nodded once, rain pouring from the battered brim of his hat. "I know who you are."

"Please identify yourself." He could see me in his monitor, could see my hands resting lightly on the trigger grips of my pocket lasers. That threat gives me a psychological edge when questioning suspects. Seeing the burn down triggers makes the threat more real to them.

No fear reaction in close up video scan of his face. But there was something so strange about this old man that I found my own face tightening a little. I found my hands sweating on the trigger grips.

"Have you seen a man on this street? Did a man pass through here tonight?" asked the old man.

Stunned, I automatically shook my head no.

"Was you here last night?" Did you see a man come through here last night? Did you see a man here after curfew?"

"Hartmann here." Audio cut in. "Play along with him. Keep him talking. We've punched in voice prints, visual factors. We're running everything through the mill again.

We have to have a computer error somewhere, possibly a circuit breakdown."

"I was on duty last night. I saw several men but none after curfew. We had a woman after curfew but no men." I answered, beeping in an affirmative to Commander Hartmann's call.

The old man's eyes burned in my monitor. The old man may have looked like a corpse but there was something fierce and wild about his eyes. They seemed to look right through me.

"Who are you looking for? Perhaps I can check with Central and locate him for you?"

He shook his head.

"I could send out a mobile unit to locate him for you."

"I'll find him first. I don't need you to find him. I'll find him first and then . . . . . "He let the sentence trail off.

"Does this friend of yours have a name?" I asked, trying an indirect tack. If we could pin down an associate, maybe we could trace back to him.

"He's no friend of mine!" snarled the old man, an edge of violence in his voice. "I've got a *message* for him."

There was an unspoken threat in his manner, in the way he emphasized the word message.

"Perhaps we could help you deliver the message." I volunteered.

"No! Not yet. The only message I got for him is under my coat."

He tapped one of the bulky pockets of his overcoat.

I punched into the mobile unit, x-rayed him, scanned him with a metal detector. The unmistakable outline of a knife came from the pocket he had tapped with his unmanacled hand.

I debated immediate confiscation but tabled it. As long as he was talking, and since he was manacled to the mobile unit and couldn't go anywhere, there was no sense in taking any overt action that might make him stop communicating. Nothing forced here, just playing along, hoping he would give out some useful information.

Central punched in again. "Hartmann here. There is no, repeat, *NO* record of this man anywhere!" There was a note of panic in his voice. I could tell he was shook up and I didn't blame him. A contradiction like this could disrupt our entire society.

I wiped my hands against the armrests of my womb couch. I was sweating like a bandit cought on a monitor! At least, this night wasn't boring any more, I'll say that much. It was turning out to be one hell of a strange night.

The old man looked back over his shoulder again. He seemed to be waiting for someone.

I piped into Central with a query. "No possibility of programmer error?"

Hartmann punched right back. "None! We've checked and double-checked! We've got a file on every living human being! We've got everyone but him!" In the background of Hartmann's signal, I heard the sound of voices in heated argument.

"Who are you? Please identify yourself!" I asked again, at Hartmann's insistent urging.

To my surprise, he told me.

"My name's Farris. Jonathan Farris." Again the old man looked back the way he had come and shivered in the rain. He was cold and wet and miserable. If there hadn't been something so wrong with him, so evil, I guess is the word I'm looking for, maybe I would have felt a little bit of pity for the old man. But there was something very much wrong with this old man, something terrible and grim which stopped any pity I might have felt toward him. Besides, I'm a wombcop. I don't have much pity for anything or anybody.

"Shall I bring him in?" I queried Central.

Before I got an answer the old man spoke again.

"Bantam is his name. Michael Bantam is the one I'm looking for. He's behind me. I'm sure. I might have passed him in the rain but he'll be along."

"Checking on Bantam." clacked my computer linkup.

"I've got to meet him. You've got to let me go." said the old man, shaking his manacled hand. "I'll be late and I mustn't be late." A shadow of worry moved across his haggard face.

"But . . ." I started to say through the mobile speakers.

"Release him immediately!" Hartmann's terse command snapped across the relays. "Have him followed! We want a record of everyone he meets, file tapes on everything he does or says!"

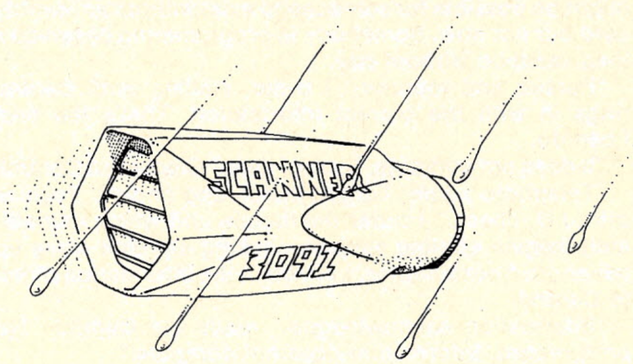
My hand jumped off the console board, curling into a fist with shock. I was stunned by the command, contrary to everything I had ever been taught. I've never let a violator go free! Not once in ten years! Not once!

"Damn it, Davis! That's a direct order! Snap to it!"

I shook myself into action, punched in the release command. I had a sick sensation in the pit of my stomach as my fingers tapped in the order. This was contrary to everything I stood for, everything I believed in.

The manacle automatically came unsnapped. The old man nodded his head and backed away from the mobile unit, massaging his free wrist.

"At least, let me confiscate the illegal concealed weapon?" I asked Central. "My God, I can't let . . . . ."



"Denied." Central's reply was immediate.

"You're free to go." I heard myself say. My hands shook on the console and I fought with myself to keep from automatically reaching for the laser triggers. My mind was crying for a burn down. My trigger fingers twitched instinctively.

"I've got to get going. He'll be coming along and I've got to find him." said the old man, touching his overcoat pocket." If you see him, you tell him that Johnathan Farris is going to get him. I'll see him killed for what he did to me."

"What does he look like? How will I know him when I see him?" I asked.

On a monitor beside my head a series of telephoto stills of Michael Bantam appeared on the screen, piped in direct from Central. As the series of photographs flashed across the screen, biographical information automatically printed out across the bottom half of the screen. Central's computers were really on the ball.

"You'll know him when you see him." said the old man with a smile that had no smile to it. "He's young, red hair cut short. There's a scar over his left eye and his face is pale like dirty newspaper. You'll know him when you see him. He'll be coming along grinning, he'll be laughing at me but not for long." Again the old man let his hand rest meaningfully on his overcoat pocket.

"If I see him, I'll tell him you're looking for him." I assured him. I glanced at the monitors. A pretty accurate description the old man gave. At least, there was a record of Michael Bantam.

There was a note of panic in his voice. I could tell he was shook up and I didn't blame him. A contradiction like this could disrupt our entire society.

Why the hell am I letting him go? What the hell is going down at Central? Have they gone soft in their computer programs? I slammed my fist down on the console, punching in angrily to Central. I'm going to get some answers! I've had about all I can take. I don't know what the hell is going on. This man's a criminal whether he's on file or not, and I got every right to burn him down.

I start to speak but the old man cut in and I listen and wait, choking on my own anger and frustration.

"He'll never get away with it! Nobody does that to me and gets away with it! I'll see him dead before the night is gone." The old man was livid with rage.

The circuit monitoring panels were all flashing emergency reds and I knew the computer system was pushing towards an overload.

I punched a sharp query at Central. "What the God Damn . . ."

"Why don't you follow me?" said the old man, beckoning the mobile unit toward him. "Just down this street and left a little ways down the alley. Yes, why don't you follow me?" He began walking.

I looked at my sector chart. The alley was the cut off point at the end of my patrol sector. That was someone else's territory. I punched in this information. Awaited a go ahead.

"Hartmann here. Ignore boundaries. Follow without restriction or limitation. Full monitoring, automatic filing, total surveillance."

I shrugged. It was a day for breaking the rules. I activated the mobile unit and it began tracking and pursuing the old man. Together, they moved down the street toward the alley.

I started to beep in an acknowledgement of the order.

Suddenly, everything went dead. Console, monitors, linkups, activation circuits. Everything. Nothing coming in, nothing going out. Computer overload. It had to be. The existence of the old man with no identity records, with no file tapes was an insoluble problem. It wasn't supposed to be possible.

The womb couch cradled me like a hand, the release catches that would free me from its comfortable grip, frozen into place by the power failure. I sat in the dark, felt like a helpless stuffed animal in the hands of a child.

I never felt so useless in my life. I struggled against the lock in the couch web, trying to force it manually, but it was impossible to shake loose. I was stuck there, helpless, like a butterfly stuck to a display board with a pin.

I shouted my frustrations at the darkened console in front of me. There was nothing I could do but wait. Nothing, not one damn thing!

It wasn't a minor overload. It must have been the granddaddy of granddaddies. My entire sector, from street unit to computer master terminal had blanked. Whoever was responsible for programming a computer solution on this case ought to get burned down. It was an error on the scale of programming a computer to find the square root of zero! Somebody was going to be up the computer without a program!

There must have been one hell of a lot of damage to repair. My wrist chronometer wasn't working. Just guessing, I'd say I sat there maybe an hour or more. Probably closer to two.

The power came back on around 0418 hours. Maybe 0419.

Central was on the line while I was still blinking my eyes, trying to adjust to the console lights when they flashed back on.

Commander Hartmann's voice almost broke my ear drums. I winced under my audio helmet and turned down the audio pickup.

"WHAT'S HAPPENING DOWN THERE?" he demanded.

I rubbed my eyes, waiting for them to adjust. The monitors were flashing back on, focusing and re-tuning for maximum image clarity.

"Locate Pickup 27, Monitor 7." I shouted. The monitor for 7 had not focused properly yet. The blurred pattern on the monitor, merged and then refocused. The forward progress of the mobile unit that had been assigned to the suspect had been stopped dead in its tracks just as it was turning into the alley. When the power surged on, the unit completed the turn, its scanners probing the alley.

"Position." clacked the computer. "Pickup 27, Monitor 7."

Mobile unit moves forward into the alley, scanners set. Audio punched in.

Tapes filing. Red flash on my console. Mobile unit activates an emergency panel. Other units from other sectors on standby with possible intercept patterns.

There's a body in the center of the alley. My mouth drops open in shock. The computer frantically begins absorbing data, counter-referencing, automatic alert all sectors.

That haggard face, the sunken eyes, the old coat. A knife sticking out of the old man's chest. Unmistakable.

I go to full zoom, extreme close-up, lateral pan. Very clearly marked. A color coded homicide tag attached to the handle of the knife. I punch in for a close-up on the card. It tells me that the victim was murdered, unmonitored, discovered by first shift of sector eight, assignment G, shift one carry-over, that the body was overdue for pick up by sanitation. There was a blue sticker on the end of the tag that meant preserve body for evidence, autopsy mandatory.

Sweet Jesus!

The computer reads out. "DECEDENT . . . FARRIS, JOHNATHAN FRANKLIN, MALE. CAUCASIAN. AGE 57. BIRTHDATE 2053/03.09. CAUSATION: Knife wound through right ventricle. ESTIMATED TIME EXPIRATION . . . 3 hours, 27 minutes, 55 seconds when first discovered. UPDATE EST. T. EXP. this scan: 6 hours, 19 minutes, 31 seconds. DEATH . . . instantaneous. CONCLUSION . . . HOMICIDE. MOTIVES . . . UNKNOWN. SUSPECTS . . . UNKNOWN. Actual crime unmonitored. No more information available without request through proper channels to sector 8. Case jurisdiction . . . sector 8. System breakdown, factor in loss of information. Suggest alternate . . ."

I cut the computer off and sat on the switch that hooked me into Central. Commander Hartmann appeared on a video monitor to my right. My console camera automatically plugged me into his office.

We just sat there and stared at each other, too shocked to even speak. I felt sick, physically sick.

"When a man dies, they take his identity file off record," said Commander Hartmann. His face was pale with shock. "The computer was able to correctly identify Farris . . ."

"But . . ." I started to say.

"From information already on file in the Death Register." He continued.

I got a cold feeling in the pit of my stomach.

"Are you trying to tell me the reason we couldn't get a make on him is because he was already dead! Are you saying he was dead when I picked him up in my sector? That we had no tape records of him because his files listed him as deceased?"

Commander Hartmann shuddered and stared down at the blank surface of his desk. "I don't have an explanation. I'm not sure I want one. Christ! Christ!" A nervous tic jerked one side of his face grotesquely. He was struggling to maintain his grip on reality.

Jesus! I turned away from Hartmann's monitor and stared at the corpse of Farris.

"He was dead three hours before I punced into my shift! But . . . but . . ." Words failed me. I couldn't move, couldn't think. I sat in my womb couch, paralyzed.

I'm just a womb cop, an extension of my computers, the driver of the car. My job's driving, punching in and doing what I'm trained for, not explaining the engine. This was out of my league. I only know what the computers know. Then I act on it. That's my job. That's all I want to do.

Commander Hartmann was on the edge of hysteria.

"Command decision!" he ordered, his voice ragged. It was a direct order.

I was confused. Hell, I was scared. I was terrified. I knew the decision he expected me to make. I just sat there stunned. I wanted to pretend I couldn't hear him, to pretend that I didn't know what the hell he was talking about.

"I SAID COMMAND DECISION!" repeated Hartmann shouting, his voice cracking with emotion.

I punched into Central, pressed the automatic filing code. I tried to stay calm but my hands shook as I dialed in.

It was the hardest thing I ever did, the most difficult command decision I ever made.

My voice sounded distant and cold, as if it belonged to someone else as I punched in the only command decision I could possibly make. "ARREST MICHAEL BANTAM FOR THE MURDER OF JOHNATHAN FARRIS. CAPTURE AND EXECUTE ON SIGHT. VIOLATION OF CRIMINAL CODE, SECTION 81-4. THIS IS A PRIORITY COMMAND. IMPLEMENT IMMEDIATELY."

The report goes to Central. The first time, I hope the only time in my life, I've made a decision that isn't based on cold hard facts. You tell me what my decision was based on. An eyewitness account of a murder from the victim? I'm not sure I know.

I waited for a decision from Central. They have all the evidence I have. Trouble is, the information I acted on, will never compute, and I know it. They could have my head for a thing like that. I'm a mass of jelly, a shock cube of raw nerves, waiting, just waiting.

The seconds crawl by slowly. I can feel the sweat pouring from me, seeping into the soft cushion of the womb couch at my back. An hour goes by.

Central links up. "SUBJECT: COMMAND DECISION OF WOMBCOP 345-45 STEVENS, ROGER DAVIS, CASE NUMBER 87-411a (SECTOR 8, JURISDICTION SUPERCEDED, APPROVED TRANSFER COMMAND DECISION TO SECTOR 7) HOMICIDE, DECISION ON APPREHENSION AND EXECUTION OF MICHAEL BANTAM . . ."

There was a pause. Oh God no, I thought, here it comes . . .

Central continued, "APPROVED. MICHAEL BANTAM APPREHENDED DISTRICT 9. EXECUTED FOR VIOLATION OF CRIMINAL CODE, SECTION 81-4. CONFIRMED. SPECIAL CITATION OF MERIT ISSUED THIS DATE, WOMBCOP 345-45 STEVENS, ROGER DAVIS FOR INDIVIDUAL EFFORT WITHOUT AID OF COMPUTER ASSISTANCE . . . CONGRATULATIONS."

What really happened that rainy night? I'm not sure I really want to know. As a very good computer friend of mine once said, it simply does not compute.

Down deep, I kind of hope it stays that way.

There are some things, that computers are not meant to know. ■

INSPIRATION

Electrons have minds of
Their own.
There is no lobotomy
For them.
There is no shock treatment
Short of a Cyclotron, and
Even that is not permanent.

In a bundle of electrons,
Some are tied to atoms
And some are not,
They are merely
Running around the brain,
Electro-Chemical impulses
Which now and then
Pause long enough to become
Poems.

Esther Gloe

AN INSTANT

Each word touches
The ear. Its
Image rests on
The brain. For
An instant only
The signal is there,
Not strong enough
To kill.

It is like lightning
which strikes
A power line,
Making sparks, but
Not burning through.
It is not on the wire
Long enough.

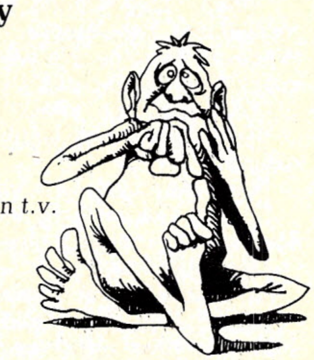
But, if lightning
Strikes a line
Which leads to
The computer,
A flip-flop will
Sense it, and
Toggle.

Esther Gloe

cosmic rays are bombarding my body

Cosmic rays
are bombarding
my body, streaking
through my flesh at
relativistic speeds,
yet I sit here fearlessly,
watching the Super Bowl on t.v.

Steelers 21
Cowboys 17



Peter Payack

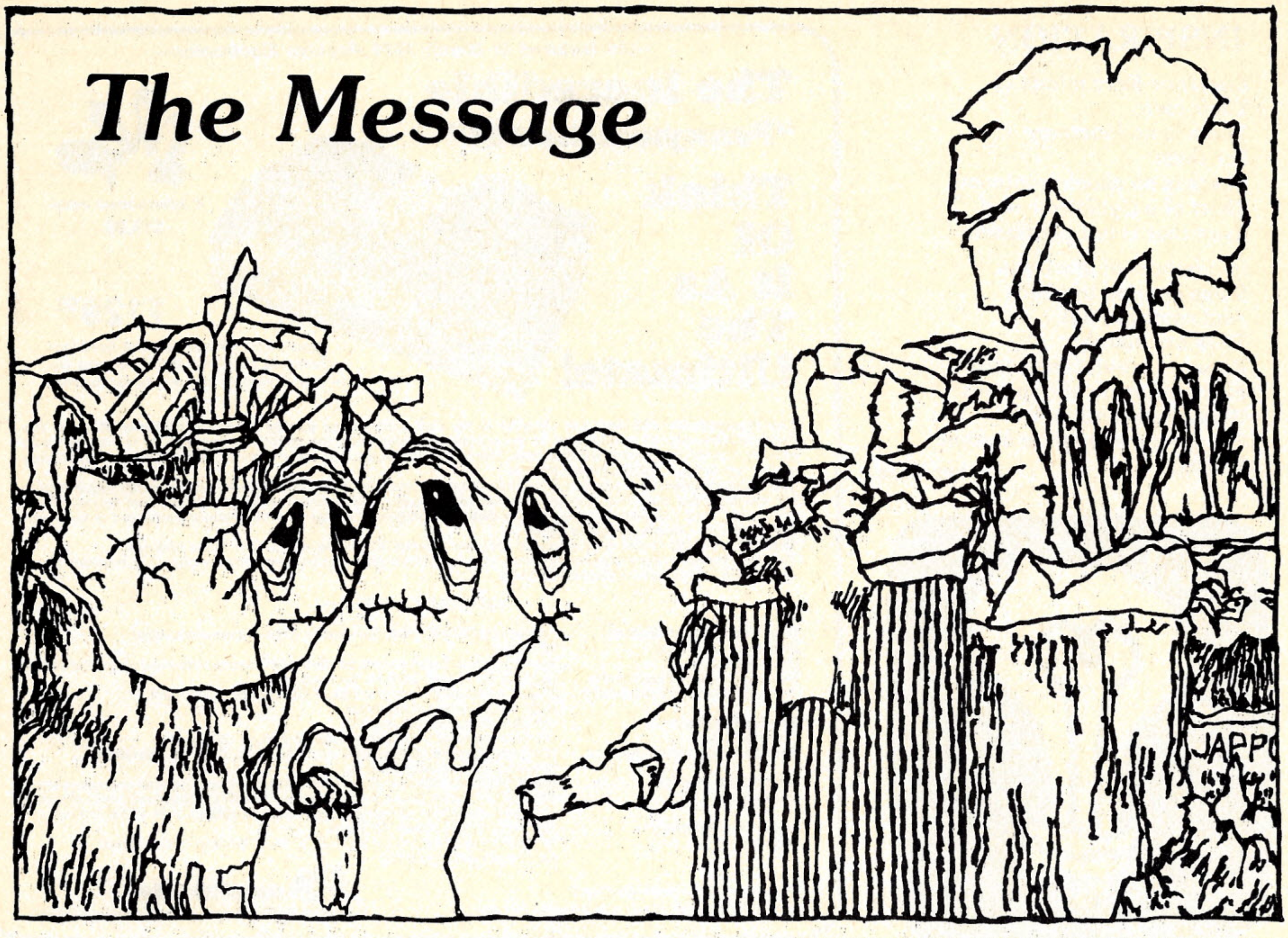
Poems by Peter Payack have appeared in *The Paris Review, The Village Voice, Darkhorse,* and *Star-Web Paper.* A collection of 14 of Peter's poems are available in the booklet "Cornucopia" available for 70¢ from Peter Payack, 23 7th St., Cambridge, MA 02141.

PL/I VALENTINE

by Jane Barnes

```
     Dcl:            my heart (
  internal, fl    oating, *); on
 overflow it go   es to where_you_
are, whose attributes recursively a-
lign, my data base to yours, by love
 on-line.  Dcl: affection (simple,
  real, defined)  area (99,999);
   although  signed my declar
    ation is refined; if you
      return heart=mine then
       go to B;  procedure
        CISL,     entry
          B:  my val
             entin
               e
```

Note:This poem was sent to the mailboxes of all on-line users on Valentine's Day, and automatically printed when they gave the command to read their mail. It was written on Multics at MIT.



ILLUSTRATIONS BY SUNSTONE GRAPHICS

*Lincoln Stein**

The earth was dead. There was no movement on its surface that could have been attributed to life. The seas were empty of even the simplest organisms. The skies were barren. The teeming of small creatures underground, the sounds of animal activity, the varying CO_2 level of seasonal plant life, these were all gone.

On that very sphere where a proud race had reached for their dreams, now existed only wind, dust, and relics of a lost world.

Then the Aliens came. One by one their ships landed in the center of what once had been a great city of men. With their indescribable bodies, carrying indescribable tools, they began their work. The aliens waded through the rubble of a bygone civilization. They kicked and scattered the dreams of a world. They defiled the sacrosanct temples of a culture. They befouled the remnants of the past.

Eep said to Ogg, "I wonder what they did with this?" He nibbled the rim off a Goodyear tire.

*Lincoln Stein wrote this story while he was in the ninth grade, at his home in Pound Ridge, New York.

"Any good?" asked Ogg.

"Needs Magnesium."

"Yerch," Ogg intoned, slithing his vorces between two mesculated pseudopods, "That's like everything here. Whoever these people were, they certainly weren't good cooks. Okk! Maybe that's why they left!"

"Yich, yich, yich, yich..."

"Hey, what are you laghing at?" Ogg's vorces twitched in irritation.

"Yich, yich, yich, yich, yich..."

"I think it's a perfectly sensible idea and—and I'm gonna suggest it at tonight's meeting."

"You're what?!"

I'm going to bring it up at tonight's meeting."

"Listen, Ogg, you bring *that* suggestion up and you'll get yourself thrown out on your sinolds, not to mention being the yiching stock of the team. If I were you, I'd just keep real quiet and not say a word, that way you'll never get into trouble.

"Besides, how do you know they left? Maybe they just all died out."

"Without leaving a trace," Ogg asked. "The anthropologists say..."

Eep was just sitting down on a box of dental floss when a shrill scream pierced out from behind one of the diaper-pin factories.

"That's Ent!" shouted one of the floogles.

"We've got to save her!" Another shrieked.

"Why should we? She scratched and consumed too much cat food." Pipsqueak piped, but he was ignored.

Captain Arnoff began giving orders, "Quick, follow me, she must be in some terrible danger. You there, and you, get the thribeams, and hurry!"

The twenty Floogles did their best to get to Ent as quickly as possible, but the going was maddeningly slow. They had to scramble over immense piles of rubber bands, wad through stacks of pamphlets on "Transcendental Meditation" "How to Stay Young and Beautiful," flounder through pits of false eyelashes and ball bearings, hazard their way through obstacle courses of useless dishwashers, keep their balance on treacherous fields of computer-confetti, and still head towards the ever-weakening shriek.

Finally they came through a wilderness of magnetic tape to find, not the Ent-Devouring monster they had expected, but something almost as frightening. In the middle of a circular clearing there was a tunnel big enough to fly a ship down. It was walled with a bluish, glassy material that glowed fitfully in the sunlight. It was from this tunnel that Ent's shrieks issued.

A few Floogles would not enter the tunnel, pleading exhaustion, and knowing in their disgrace that everyone else knew that they secretly feared the ominous tunnel mouth—so utterly unlike anything else they had come across in their explorations.

Not a few of those who were proceeding down the tunnel were also trembling, but devotion to Ent and loyalty to their captain kept them moving forward. Perhaps the only one whose mesculated pseudopods did not shake was Captain Arnoff, a floogle long admired for his bravery. He led the rescue team down the long tunnel as its polished sides began to glow more intensely.

As the glow became increasingly apparent, the Floogles mesculated pseudopods began to shake with an ever increasing tremor, for floogles, though inhuman, are no strangers to human fear.

The glow grew bluer, and began to ripple, but the tunnel's curving sides blocked any view of the glow's source. The plaintive cries of Ent had long since been stilled, a fact which made even Captain Arnoff feel a sense of impending doom. But it was too late to turn back, the steps they took were as inexorable and irretrievable as time itself, and their paths were for evermore frozen. They could no more turn back than alter the fate of the planet they were exploring. And now, from the brightness of the light, they could see that they were approaching its spring, where they might learn the answers to all their convulated questions, and lose their lives in so doing.

They entered a circular chamber, now as speechless as the day they were hatched, and in the grip of inexorable power, beheld the summit of that past civilization's dreams. In the center of that chamber was a machine, gigantic, godlike in its every aspect, seeming to possess life in a way never imagined. It burned itself upon the Floogle's image receptors so that all else seemed to recede and vanish. They froze in time and space. Even Ent, the ship's cat, was hypnotized in this way.

As the Floogles helpless watched, the glow emanating from the machine's reflecting/refracting sides became intensified, changing its hue in some indiscernable way from ultra-marine to blue, to dark green to light green to yellow to red, rust red, and then blood red, flowing and seeping into every nook, crack, corner and cranny of the chamber—into the very minds and bodies of its occupants.

The glow thickened for awhile. Then a low, almost imperceptable hum awoke in the depths of a machine that had lain inactive for millenia, till it could fulfill the purpose for which it had been set upon the Earth.

Finally, with a soundless grinding of non-existant gears and cogs, the machine spoke. In a low monotonous voice that droned on and on, it delivered its age-old message. What it said was this:

"For millions and millions of years I have awaited some visitor from the heavens that I might deliver the message and carry out the program that was given unto me at the passing my masters, millenia ago. They realized that their world and civilization were failing, so they wrote one last great computer program, and gave it in trust to the last great computer, that darkness not fall forever. This program enables me to analyze the fall and warn you and others who may follow you away from the maelstrom that hurled my makers and this planet into oblivion.

"The RUN cycle will commence in 5 seconds."

The great computer whirred, and then said:

"ANAL

"UNDEFINED VALUE ACCESSED IN LINE 2315

"UNMATCHED NEXT IN 3473

"SUBSCRIPT OVERFLOW IN 4090

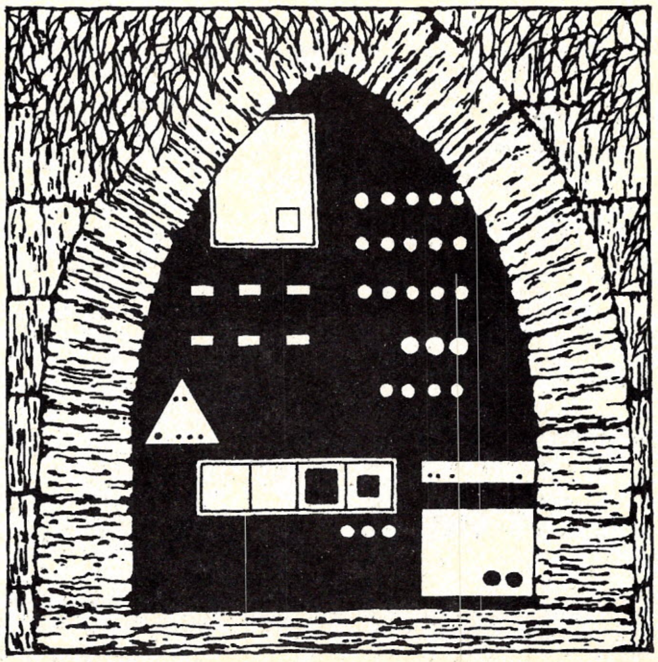
"MISMATCHED VARIABLE IN 4200

"DATA OF WRONG TYPE IN 6045"

The machine, having run the program fed into it so many long ages ago, gave up the last semblance of life and ceased functioning. The light slowly dimmed, and Earth was silent at last.

The Floogles, as if released from a spell, stood up and shook themselves. They collected Ent who apparently had dozed, and set back to base. On their way, they discussed the message and the strange "program," deciding that though it was mysterious, it wasn't surprising for "earthlings." Several days later they left for their home planet and never returned to Earth. Their account of Earth, the machine, and its mysterious message were eventually misfiled.

MORAL: Garbage in—garbage out. ■



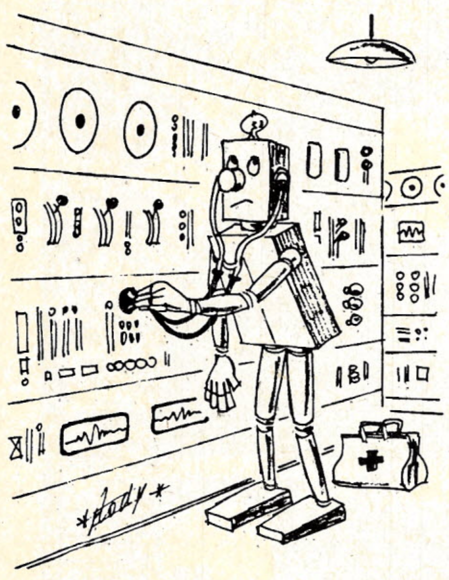
First of a new series on logic in its many applications.

Topics in Logic

John Lees

This is the first in what I intend to be a series of articles on all those topics which fall under the general heading of Logic. Instead of a sequence of articles going from "simple" to "advanced" logic, I am going to treat individual topics which more or less stand alone as complete articles. I will try to treat topics of current interest and those which, while usually embedded in esoteric books or advanced courses, are of general interest and use.

A wide area is covered by the term "logic." Since I have a background in both philosophy and computer science, I intend to cover quite a range of topics, from syllogisms to predicate calculus, digital-circuit logic, computer-aided design, computer languages, natural-language processing, parsing, compiling, formal grammars, special-purpose languages such as Lisp, Pascal, and Smalltalk, Turing machines, finite automata probability, many things from the area of artificial intelligence, the ethics of machines, artificial human languages such as Loglan, the history of logic devices, the inadequacy of Asimov's "Three Laws of Robotics," and much more. Some of these topics will be covered in-depth, some in overview. If you have preferences or other ideas, let me know.



Now available! The colossal Book of Computer Cartoons—$5.95 postpaid ($6.95 foreign) from Creative Computing, P.O. Box 789-M, Morristown, N.J. 07960.

What is a Computer Language?

If one really wanted to do so, logic could be divided into three areas: stuff that is so elementary and fundamental that it doesn't apply directly to the "real" world, stuff that at the moment is actually good for something, and stuff that will come in handy someday but right now appears worthless. I'm going to begin with a topic from the middle category, the "applied stuff" category which contains many of the tools of computer science. Since there is so much discussion today about which is the "best" computer language and about "tiny" and special-purpose languages, let's take a look at just what a computer language is, and is not.

Language is one of those concepts which are not easy to define. Everyone will agree that language has something to do with communication, but opinions differ on what makes one language better than, or even truly different from, another language, or on what effect a language has on the way people view the world or attempt to solve a programming problem. Human languages are usually rather loose affairs, but there does exist a way to formally define a language so that it may be manipulated with the tools of logic. Much of computer science depends on this ability.

A language consists of three parts: A set of symbols (an alphabet) which may be used to construct strings; the set of permitted, or correct, strings; and the meaning of the permitted strings. Thus a language consists of *symbols, syntax,* and *semantics;* change any one of the three and you have a different language. In practice, closely related languages are usually called dialects. Cobol and Fortran are different languages, but Watfiv is a dialect of Fortran.

The symbols of a written computer language present the least problem. Usually the matter is settled by looking at a keypunch or Teletype keyboard. Some languages, such a APL, use inconvenient methods to expand the paltry and inadequate set of symbols offered by the standard keyboards to get around the untold confusion that has resulted from, for example, the practice of using the symbol "=" as an assignment operator when it was already well defined as an equivalence operator in mathematics. A few very graphics-oriented languages, such as Smalltalk, where one may use turtles and pointing fingers and Old English script if the mood strikes you, are hopefully showing the way to the future.

The syntax of a language is nothing more nor less than a definition of the ways in which symbols may be grouped to form strings. In this context a "string" may be a single symbol, a sentence or statement, or an entire program. The usual manner of specifying the allowed strings of a language is with a grammar. Briefly, a grammar is a set of rules that allow you to tell if a particular string is in the language. Since most interesting languages have an infinite number of correct strings, simply listing the strings is not practical.

After the set of symbols, which we usually take for granted, syntax is the most obvious attribute of a language. Syntax tells us that GOTO 5 is a proper string in the Basic language whereas 5 TO GO is not. One of the first things a compiler or interpreter must do when given a program or statement is to determine if it is syntactically correct. Only then is it possible to go on to semantic analysis and set things up to carry out the meaning of the program or statement.

The semantics of a language, and the semantic analysis portion of a compiler, is the trickiest and least understood aspect of language design. Here it is where we must decide the exact meaning of a statement such as GOTO 5. It's transfer of control but...can you transfer out of a loop, out of a subroutine, are any variables changed during a transfer, can a statement transfer to

itself, can you transfer to a non-executable statement? Does ONE+TWO mean addition, concatenation, or the Boolean OR? There are many such questions of semantics which must be settled and rigidly defined before a language is ready to be loosed on the world.

An added complication to defining syntax, semantics, and even the symbols of a language is that they tend to change slightly with implementation and hardware. Such things as size of line numbers, accuracy of numerical calculations, maximum length of character strings, speed of execution, and effect of carriage-control characters in print statements must be watched very carefully. Computer languages are still a long way from being machine-independent. Pick any popular language and you'll quickly run out of fingers trying to count the different versions, improvements, extended supersets, and simplified subsets. It's a wonder that it's possible to write *any* portable software!

The problem of lack of standardization in computer languages is a serious one. Languages are the basic tools of the trade for people who design and write software packages and application programs. There is a lot to be said for being sure of your tools. Cobol and Fortran are the most popular languages not because they have no deficiencies, but because they are supported in most of the world's computer installations Having a firm mastery of a computer language is a lot like having a firm mastery of English: you are free to concentrate on what you want to say, instead of having to worry about whether adjectives come before or after nouns.

On the other hand, the people who say that languages such as Cobol and Fortran are archaic monstrosities that encourage poor programming and loose thinking have their points. Structured programming in Fortran is almost meaningless, and debugging a large Fortran program can be really close to impossible. PL/I, meant to be a universal language, is almost too complicated to completely master and frequently ensnares programmers who try to use its neat features to take "shortcuts." Everyone who does any programming at all eventually comes to feel that there must be a better way to do some things, and some of these people design new languages.

For the hobby-computer user, the most pressing reason to design or use something other than a "standard" language is the limitations imposed by the available equipment. Fortran simply is not going to run in 4K, no matter how clever a programmer you are! Most of the users of "tiny" languages are concerned with having the most programming power possible with the equipment they can afford. There is also a certain satisfaction that goes along with successfully programming in a minimal language, much akin to radio amateurs who go the QRP route. After all, since a computer is a universal Turning machine, it should be possible to eventually do anything with any language.

The commercial and scientific computing world has long since stopped worrying about how much core a language compiler requires, rather the stress is on features of a language which aid good programming practice or provide special capabilities such as matrix manipulation, list processing, or complicated data structures. Designing, implementing, and debugging a new commercial language is a very costly and time-consuming process, not a task that is undertaken on a whim. Resistance to new languages is also high, most shops prefering the beast they know to a new, untried nest of problems.

This has been, obviously, a general overview of computer languages. Next time we will take a *very* simple language, look at its grammar, and see what makes it tick. ■

Personal Computing Expo will come to New York!

PCE PERSONAL COMPUTING EXPO
NEW YORK COLISEUM, OCTOBER 28, 29, 30, 1977

It's a brand new show (in the world's biggest economic center) specifically for manufacturers, buyers and those providing services to the personal computing enthusiast. For the first time, this booming field will have a New York Coliseum showcase. It is planned as the largest public show of its type and will attract attendance from the huge population areas of Boston, Washington, Philadelphia and Baltimore.

New York's is the world's communications focal point, the one place that will put personal computing in a significant spotlight. New York is surrounded in depth by people who work in the computer field, by computer learning centers, universities, personal computing clubs and thousands of others whose lives are affected by computers.

More than 100,000 paid subscribers of BYTE magazine will be urged to view the exciting exhibits and to attend the BYTE-sponsored lectures. Circle the dates: October 28, 29, 30 — and make your plans now to be there when Personal Computing Expo comes to New York!

"Personal Computing: An idea whose time has come!"*

Tutorials by the IEEE Computer Society, Mid-Eastern Area

Personal Computing Expo is also joined by the Institute of Electrical and Electronics Engineers Computer Society, Mid-Eastern Area Committee, whose experienced staff is presenting six day-long tutorials at a modest charge. If inconvenient for you to attend a tutorial during the show, simply sign up for follow-up tutorials on weekends after the show.

Whereas the lectures will provide you with information, the tutorials will teach important skills enabling you to use your own computer at home or at work.

TUITION:

The tuition fee for the tutorial program includes a one-day admission to the Personal Computing Expo.

	One Tutorial	Two Tutorials	Three Tutorials
Students (with ID)	$30	$50	$75
IEEE Members	$40	$70	$100
Non-Members	$50	$90	$125

Tuition includes hand-out material, including text and/or hand-out materials. Participants will also receive a certificate of participation.

REGISTRATION:

In order to provide an interactive, learning environment between the participants and the lecturers, the number of registrants is limited. Registration is accepted on a first-come, first served basis. Early registration is therefore suggested. Cancellations received before September 15, 1977 will receive a full refund.

To register, make your check payable to the IEEE COMPUTER SOCIETY, and mail to:

Daniel R. McGlynn, Ph.D.
Tutorial Program Chairman
IEEE Computer Society
329 - 84th Street
Brooklyn, N.Y. 11209

FOR FURTHER INFORMATION:

on the technical content of the tutorials, technical background suggested to derive maximum benefit from the program, or information on the IEEE Computer Society, call

Cary Ringel
Chapter Chairman
IEEE Computer Society (212) 460-4600

TIME AND LOCATION:

The tutorials will be held from 9 AM to 4 PM each day in the New York Coliseum, at a location to be announced and posted. Participation in the tutorials also includes a one-day admission to the exhibition area and other lecture programs.

EXPO TICKETS:

Will be sent to tutorial registrants about three or four weeks prior to the show.

BYTE The Small Systems Journal DAILY TUTORIALS AND SPONSORED LECTURES

Personal Computing Expo to be produced by H.A. Bruno & Associates

H.A. Bruno & Associates, Inc., has been prominent in the exposition and promotion fields since 1923. Highly skilled in the production and promotion of consumer and trade shows, the company currently promotes the American Energy Expo, the National Boat Show, Auto Expo/New York. Promotion assistance also is currently rendered to the National Computer Conference and the Triennial IFIPS Congress in Toronto.

The show producer has promoted successful shows in the New York Coliseum every year since the building opened in 1957. Staff personnel are thoroughly familiar with the building, its services, management and labor.

Interesting, educational exhibits of Personal Micro Computers

The lectures and tutorials are not just theory! You will be able to see a multitude of micro computers. Try out various demonstrations: fascinating games, even in color; small business applications; computer produced speech; music; color graphics; and many more. Micro computers are not only small and portable, also you will need only a "micro budget" to be able to take them home.

Show Hours and Admission

Personal Computing Expo hours are as follows:
Friday, Oct. 28 — Noon to 10 p.m.
Sat. Oct. 29 — 10 a.m. to 10 p.m.
Sunday, Oct. 30 — Noon to 7 p.m.
General Admission: $5.00 (includes free BYTE lectures) per day.
Two-day Tickets: $9.00 (advance sale only)
Three-day tickets: $13.00 (advance sale only)

General Information

You may find it advantageous to purchase two or three-day admission tickets in advance. These are available by mail only, no later than October 10, 1977. Use coupon below.

Group rates (10 or more persons) qualify for $1.00 off regular prices. Arrangements must be made by mail prior to October 10, 1977.

Special arrangements have been made if you desire to stay overnight. Our headquarters hotel, the Barbizon-Plaza, is located on Central Park South, two blocks from Columbus Circle. Single rooms available at $34.00 per night; $40.00 double, plus tax. There's a weekend plan: $22.95 daily, plus tax per person, double occupancy . . . includes breakfast (brunch on Sunday) and meal gratuities. Children under 14 in same room with parents, free.

For hotel reservations and information, call toll free (800) 223-5493. From New York State call (800) 223-5963.

For those traveling to New York by air, American Airlines offers a convenient service through arrangement with Personal Computing Expo. For information, call toll free (800) 433-1790. In Texas the number is (800) 792-1150. From the West Coast, round trip fare via American is only $227.00.

20,000 persons are expected to attend and view the more than 200 exhibits by personal computer manufacturers and retailers.

Personal Computing Expo will occupy the 4th floor of the New York Coliseum. It is located on 59th Street and Columbus Circle — the geographical center of New York City. Garage parking in the building is available.

For answers to any questions pertaining to your attendance at Personal Computing Expo, contact the Show Manager, Ralph Ianuzzi, at Area Code 212/753-4920.

Advance sale of tickets available . . . Three days $13 Two days $9 . . . General Admission $5 MAIL THIS CONVENIENT ORDER FORM NOW!

Personal Computing Expo admission is $5.00 per day. Advance reservation eliminates waiting in line. Order advance tickets with this coupon. Admission ticket includes access to exhibits, lectures and tutorials.

Please send me ________________ advance registration tickets for three days, October 28-29-30. Total cost $13.00 per person.
Please send me ________________ advance tickets for two days, October ______ and October ______. Cost is $9.00 per person.
Please send me ________________ advance tickets for one day, October ______. Cost is $5.00 per person.

Make all checks payable to PERSONAL COMPUTING EXPO, and mail to:
Personal Computing Expo, 78 East 56th Street, New York, N.Y. 10022.

Name ______________________________ Amount enclosed $____________
Address ____
City ________________________ State ______________ Zip ____________

puzzles & problems

Thinkers' Corner

by Layman E. Allen © 1976

MATHEMATICS PUZZLES

How many of the problems (a) through (f) below can be solved by forming an expression equal to the GOAL? (Suppose that each symbol below is imprinted on a disc.)

The expression must use:

(1) only single digits combined with operators,
(2) all of the discs in the REQUIRED column,
(3) as many of the discs in PERMITTED as you wish, and
(4) at most one of the discs in RESOURCES may be used.

Special Rules

The "*" indicates "to the power of". Thus 3*2 = 3^2 = 9.

The 'V' indicates "the nth root of". Thus 3V8 = 2.

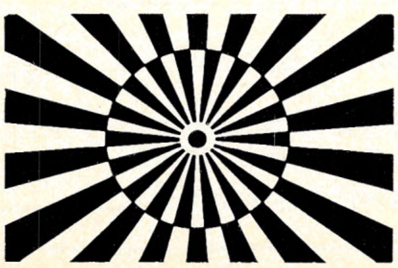
Parentheses can be inserted anywhere to indicate grouping, but never to indicate multiplication.

Problem	Goal	Required	Permitted	Resources
(a)	9	8 +	2 3 4 ÷	− ÷ V 1 3 5 8
(b)	26	4 6 x	5 x ÷	+ − x 0 1 3 5
(c)	1	2 4	1 2 ÷	+ − ÷ 6 7 8 9
(d)	7	9 −	3 5 +	− x ÷ V 2 4 6
(e)	1	4 6 ÷	1 8 +	− x ÷ V 1 3 6
(f)	9	7 4 V	4 +	+ x ÷ V 1 2 3

Some suggested answers (frequently there are others):

(a) (8 − 2) + 3
(b) (5 x 4) + 6
(c) (4 ÷ 2) − 1
(d) 3 − (5 − 9)
(e) (1 ÷ 4) + (6 ÷ 8)
(f) (2 V 4) + 7

If you enjoy this kind of puzzle, you might like playing EQUATIONS: The Game of Creative Mathematics. Free information about this and other instructional games is available upon request from The Foundation for the Enhancement of Human Intelligence, 1900-E Packard Road, Ann Arbor, MI 48104.



An Interesting Problem

by Fred Gruenberger

A financial institution advertises:

> Send us $100 per month for 12 years and we'll send you $100 per month forever.

This is simply a clever way of expressing confidence that the institution involved can maintain a 6% interest rate (compounded monthly) indefinitely. At that interest rate, $100 per month for 144 months builds up a reserve of over $21,000, and $21,000 at 6% can then generate $100 per month in interest without disturbing the principal. As a matter of fact, the build-up period could be 129 months, rather than 144, and the same offer could be made (the amount available at the end of 139 months would be $20,004.84).

They could also change the offer to read:

> Send us $100 per month for 12 years and we'll send you $200 per month for the following 12 years.

Again, the same offer could be made using 139 months instead of 144.

All the above assumes constant interest rates, and the problems could be solved quickly and readily with compound interest tables.

Suppose, however, that the interest rate were not constant. Assume that a fund of exactly $20,000 is available for payout, at an initial rate of 6%, compounded monthly, but that the interest rate rises by .00008333333 per month; that is, the rate rises to 7% at the end of ten years and continues to rise at that rate. How much can be paid back per month, so that the fund is completely depleted after 139 months?

Reprinted with permission from *Popular Computing*, Vol. 4, No. 1 (P.O. Box 272, Calabasas, CA 91302)

COMPUTER RECREATIONS

by D. Van Tassel

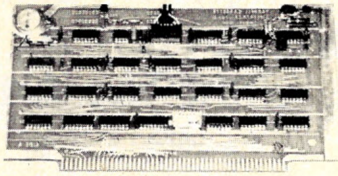
Syntax Messages

In the March-April 1977 issue I suggested you write a program to generate as many different syntax-error messages as possible with as few statements in the program as possible. I received a few responses but first prize must go to Wayne M. Compton of Dhahran, Saudi Arabia. He sent me a COBOL program with just one statement which generated 570 error messages. The statement was just the program name paragraph:

```
PROGRAM-ID. ERRMSG.
```

The compiler then went wild and generated 570 error messages. He ran this program on a IBM 370 OS/VS system. If you have access to such a system you might try it.

Why Wait?



The Tarbell Cassette Interface

- Plugs directly into your IMSAI or ALTAIR*
- Fastest transfer rate: 187 (standard) to 540 bytes/second
- Extremely Reliable — Phase encoded (self-clocking)
- 4 Extra Status Lines, 4 Extra Control Lines
- 37-page manual included
- Device Code Selectable by DIP-switch
- Capable of Generating Kansas City tapes also
- No modification required on audio cassette recorder
- Complete kit $120, Assembled $175, Manual $4

TARBELL ELECTRONICS
20620 S. Leapwood Ave., Suite P, Carson, Ca. 90746
(213) 538-4251

California residents please add 6% sales tax
*ALTAIR is a trademark/tradename of MITS, INC.

Are You *Overworked?*

Are you still doing those things that your computer should be doing for you? We are developing those utilities and subsystems that will allow you to spend more time getting to the essence of your development.

The latest product in our software line is:

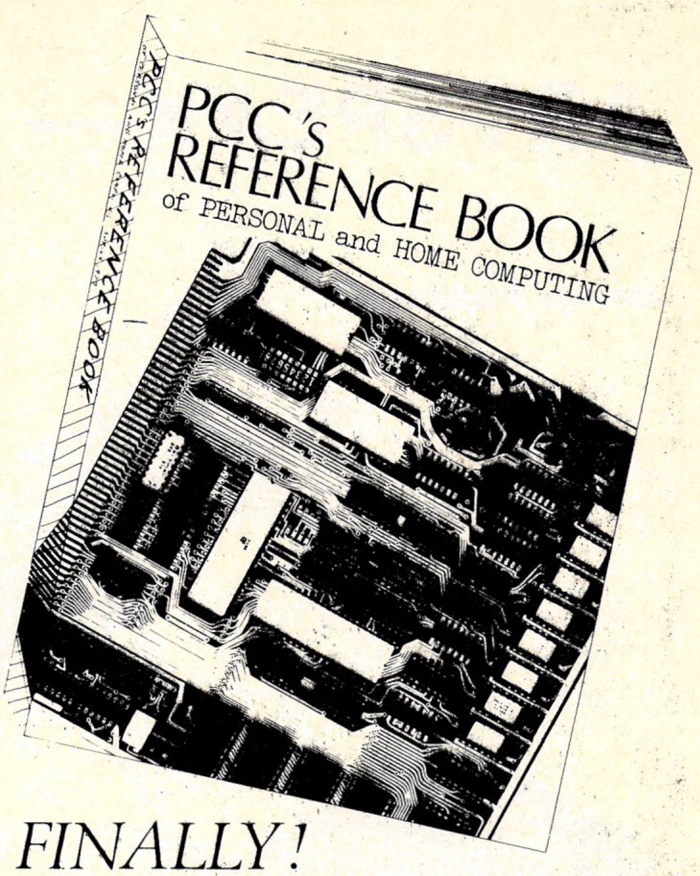
The Dynamic Debugger

Our Dynamic Debugger is a revolutionary approach to debugging. Unlike conventional debuggers, which are passive instruments that must be tediously guided to locate errors, the Dynamic Debugger is an intelligent companion which can quickly pinpoint problem areas.

Call or write for detailed specs

COMPUTER MART OF NEW JERSEY, INC.
501 Route #27
Iselin, N.J. 08830
(201) 283-0600
Store Hours: Tues. thru Sat. 10 am–6 pm
Tues. and Thurs. 'til 9 pm

THE MICROCOMPUTER PEOPLE



FINALLY!
A REFERENCE BOOK
for Home Computer Users!

Ever try to find the address of, say, a particular manufacturer when you don't even know where they are? Looking for an article on a certain kind of hardware? Frustrating, isn't it? We know, because we've been frustrated by it too, so we decided to do something about it. *PCC's Reference Book* includes:

☆ Hundreds of companies and computer stores selling hardware, software, peripherals and offering all sorts of services are listed.

☆ Nuts-and-bolts and survey articles on software, hardware, applications, robots, and the future, for the experienced and the not-so-experienced user of home computers.

☆ The complete source and object codes for a 2K Tiny BASIC

☆ A massive index of the articles from the major hobbyist magazines, plus information on magazines in the field, hobbyist clubs, newsletters and professional societies.

☆ Bibliographies on different areas so you can investigate further, with a special computer music bibliography.

You'll wonder how you ever got by without it.

Please send me *PCC's Reference Book.* My $5.95 plus 95¢ for shipping and handling in the U.S. is enclosed. (In California add 35¢ for sales tax).

NAME ____________________

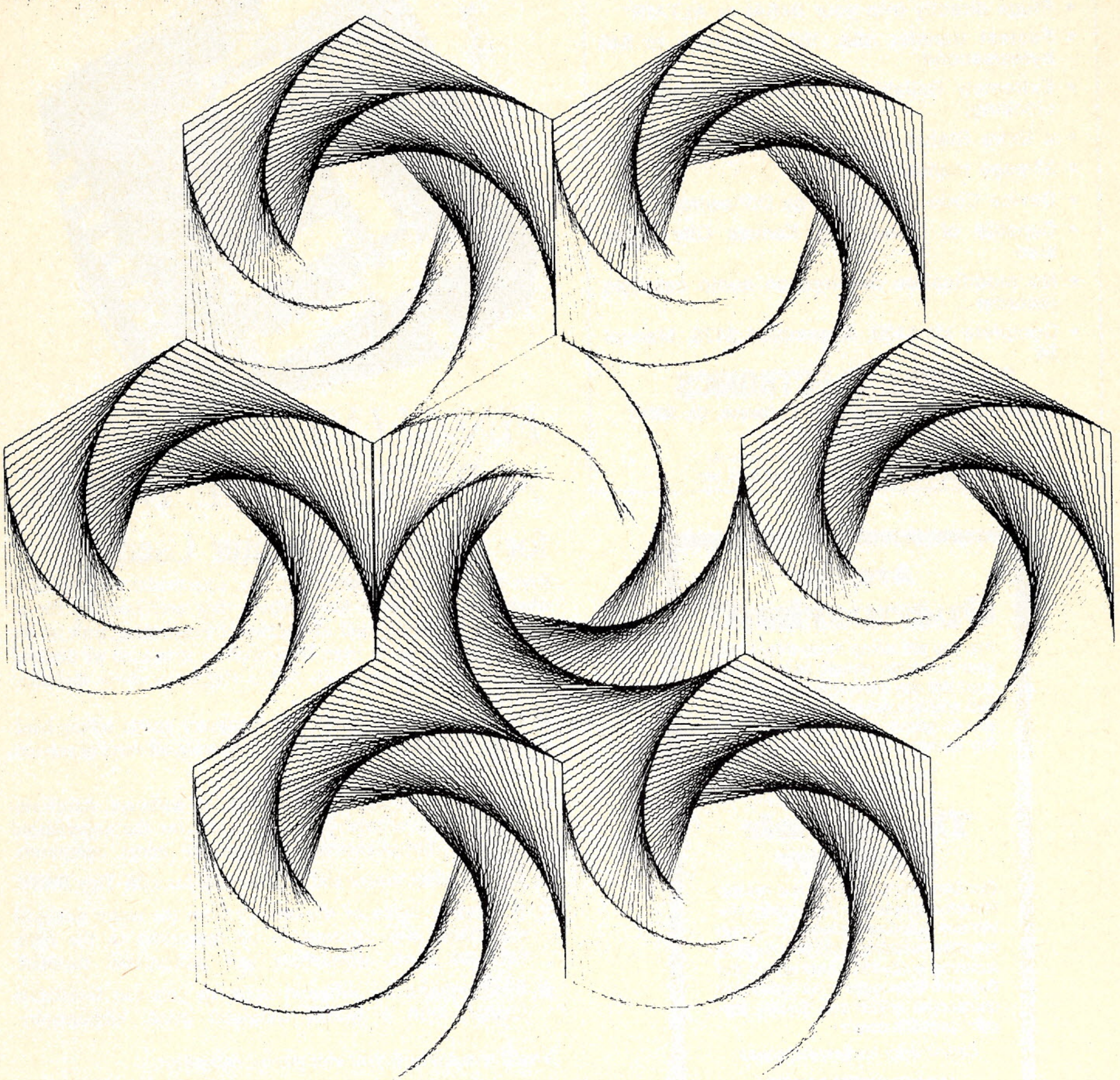
ADDRESS ____________________

____

Send to: Dept B.1263 El Camino.Box E.Menlo Pk.CA94025

ARTSPEAK — A Computer Language For Young At Heart And The Art Lover

Jehosua Friedmann*



*5 Molcho St. Rehavia
Jerusalem, Israel

In the following we intend to discuss the ARTSPEAK language, its areas of applications, the required software and hardware is then briefly outlined, followed by a programming example along with samples of output. This article is in no way a tutorial in ARTSPEAK.

What Is ARTSPEAK?

ARTSPEAK is a specialized computer language which provides graphic output. That is, the language has the potential and capability, when used properly, to output graphs, designs, curves of all kinds and an unlimited variety of original artistic drawings, some of them with stunning effects. This computer language is unique in that it has a limited syntax (grammar) and its statements are simple self documenting English statements. The language can be taught to adults and children. Artspeak is easy to learn and it can be mastered in hours.

Short History

In the last decade there was a tremendous increase in number of students taking computer courses, In particular, at present, there is a strong demand for a computer course on an exposure level like the course titled Computers and Society given in many colleges. The motivating force responsible for the generated interest was the increasing and all pervading influence of the digital computer in almost all facets of our life. At the time a child is born he becomes a number in the computer. Later, in school and in college, he uses the computer for registration, statistics, tuition payments, and even for some of his education. Upon entering adult life, his salary, federal state and city taxes, his savings and checking accounts are all controlled by a computer. When he takes a vacation, or he makes an airline reservation or perhaps rents a car he is hooked to a computer. Even when he enters a hospital he still can not escape the influence of a digital computer. In short, the computer found its way into business, military, government, educational institutions, industry and all other areas of live endeavor. Our society is by now so committed and dependent on the computer had it stopped, our national economy would cease to function.

It is generally believed that to learn about computers one must interact with them. Interaction means communicating with a computer through a computer language. However, programming (writing programs using instructions which tell a computer to perform a specific task) requires a logical approach and mind, an ability to solve problems and at least an appreciation, knowledge and in most cases an ability to tackle mathematical problems.

It is an open secret, that a large number of our high school and college student population inherited a fear towards any subject that has a smell of mathematics to it. As a consequence, teaching to students a computer language became a difficult and often an impossible job.

This problem was recognized by Dr. J. T. Schwartz of the department of Computer Science at New York University. The story so goes, at a lunch one day he outlined on the back of a table napkin the elements of a new computer language which he called ARTSPEAK. The objective was to design a language which was very simple, easy to learn, easy to write and understand, with a minimum instructions and the output was to be something pleasing to the eye.

In 1970, Mr. David Benevy, then a graduate student, (and now at Tel Aviv University) wrote a compiler for ARTSPEAK to be used on a CDC 6600. Shortly thereafter the language was improved by several students at New York University. Currently the compiler is being modified by Dr. Caroline Wardle from the Mathematics department Hunter College, New York, for use on an IBM 360/370 computer. The Artspeak compiler may be obtained by writing to the Service Bureau of Control Data Corporation, New York City. A manual describing ARTSPEAK, titled The Art Of Programming ARTSPEAK by Henry Mullish, may be obtained from the Courant Institute of Mathematical Sciences, New York University, Washington Square, New York City, N.Y.

Reaction of Users

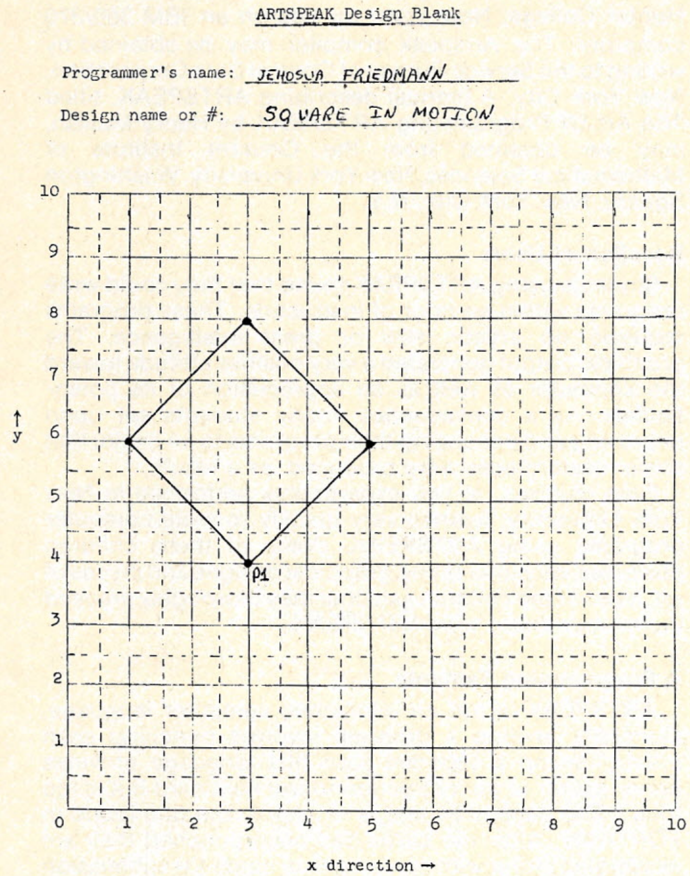
Students using ARTSPEAK found that they could write programs within an hour of instruction. These programs required no artistic skill or great intelligence. The graphical output generated a very positive attitude toward the computer as well as an appreciation of its potentialities. As a consequence, many students set upon improving and enhancing their programs to an extent that some masterpieces were produced, as a result.

Even children, at two experiments performed at New York University, learned very quickly to write computer programs using ARTSPEAK. Some of these children produced very interesting graphical designs and pieces of art. Most of the children said that they thoroughly enjoyed their experiences with ARTSPEAK.

A Programming Example

ARTSPEAK is not an interactive computer language. Generally, writing a program involves punching each instruction on a separate card (the language provides continuation of an instruction to the next card if it is too long.) The general format is similar to the one found in FORTRAN. After the cards had been punched they are assembled along with some control cards for submission to the computer center. To execute a program in ARTSPEAK the computer center must use a CDC 6600, an ARTSPEAK compiler, a card reader, disk, console terminal, printer and a plotter. After the program had been executed it is returned to the programmer along with a 10"x10" output sheet. The output includes a program listing and some output (if the ARTSPEAK compiler found no violation of the language). The programmer, then examines his output if he is not satisfied or he wants to improve his program or experiment with it, he might make changes. Usually, these changes would involve addition or deletion of one or more statements. These statements are punched on a keypunch machine and the program is resubmitted for execution.

All drawings in ARTSPEAK, that is, the output, use paper 10"x10" in size. A programming aid called Artspeak Design Blank, shown later with a sample problem, was developed to facilitate designs. This form has 10 units on the X and 10 units on the Y axis. Thus, a total of 100 squares are available for planning purposes. Each of these squares corresponds to the one inch square area on the output form. To initiate a program, one first sketches his design on the Artspeak Design Blank. Then, using this design as a guide the programmer writes his instructions, which thereafter are keypunched to form a program.



In the following we will illustrate, plan, code and explain a sample problem using ARTSPEAK.

Draw a square standing on one of its corners. Rotate the square on its corner an angle of 2° degrees and redraw the square. Repeat this process 45 times and finally stop. The ARTSPEAK program was written in two stages.

- We planned the program.

We have used the artspeak design blank form to do the planning. First, we have chosen a point with the coordinates (3,4) and marked a dot on its place. This point was labelled P1. We choose three more points (3,8), (1,8), and (3,4) again using a dot to mark their position on the form. Finally, we connected these points with lines which formed a square standing on one of its corners.

- The artspeak program-coded.

```
1            LET P1 BE POINT (3.4)
2            LET C1 BE LINE P1, (5,6), (3,8), (1,6),
             (3.4)
3    L1      DRAW C1
4    L2      ROTATE C1 ABOUT P1, ANGLE 2
5            REPEAT L1 TO L2, 45 TIMES
6            STOP
```

Explanation of the program.

Statement 1 tells the computer to define a point at 3,4. This point is labelled in the program as P1 as was also done on the design form. (In artspeak a point may be represented by using the letter P followed by any positive integer selected from 1 to 100)

Statement 2 then defines a curve which connects these points to form a square. This square is formed starting at (3,4)=P1 and connecting the points (5,6), (3,8), (1,6) and finally joining with point (3,4). The square identifies a curve labelled C1. (In artspeak curves are identified by the letter C followed by any positive integer selected from 1 to 100)

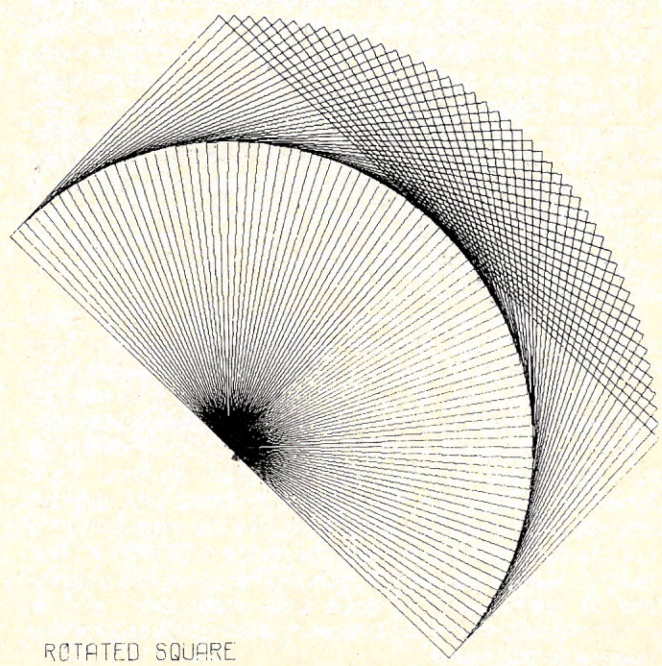
Statement 3 commands the plotter to draw the square.

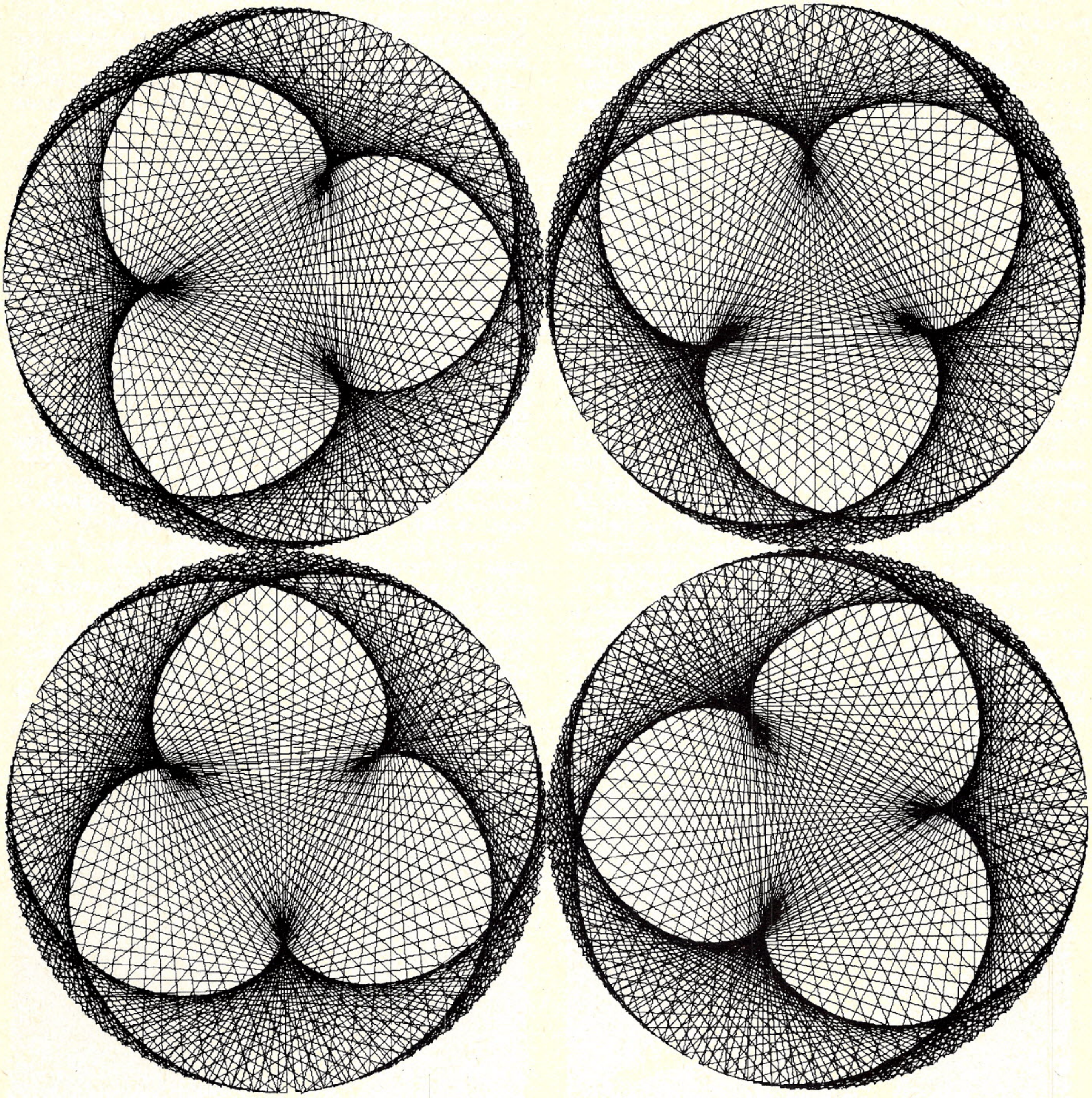
Statement 4 tells the computer to rotate the square C1 about point P1 2 degrees clockwise. (You may note that artspeak allows one to assign a label to each of its statements, however, only statements which are referenced are usually labelled. All labels must start with the letter L followed by an integer from 1 to 100. In this program we have assigned labels L1 and L2 to statements 3 and 4 respectively.)

Statement 5 tells the computer to repeat this process (steps 3 and 4 in the program) 45 times.

Statement 6 will stop execution after 45 repetitions of statement 5 had been completed. ■

The output





ARTSPEAK sample output by J. Friedmann.

BROWN SCIENTISTS PEER INTO FOURTH DIMENSION

Curt Norris*

Two Brown mathematicians have succeeded in penetrating the realm of the fourth dimension through the use of very sophisticated computer graphics techniques. The unique studies, which allow the scientists to visualize the unseeable, are conducted by Thomas F. Banchoff, associate professor of mathematics, and Charles M. Strauss, assistant professor of applied mathematics.

The two Brown University professors are among the very few people in the world who are able to manipulate pictorial representations of four-dimensional concepts. They create their art by instructing the computer to investigate geometric objects that cannot exist in our three-dimensional world. The computer then produces images on a television screen in the Brown University Computing Laboratory.

As the professors twist a "stick" resembling an aircraft control, the cage-like image rotates and tumbles like an architectural model in the grip of robot arms. Then a turn of a dial sends the geometric form through rotations in four-dimensional space, producing sweeping changes in the shape of the figure.

"Students in my mathematics courses and my freshman seminar on the Fourth Dimension say that they react entirely differently to higher dimensions after becoming familiar with these representations," Prof. Banchoff reports. "The possibilities of these techniques for mathematical research are very exciting to the scientists who have seen these demonstrations," says Prof. Strauss.

Prof. Banchoff said that a good analogy of a fourth dimensional concept would be the lifetime of Abraham Lincoln. No one three-dimensional reconstruction of a historical moment would give the complete picture of the man, but a collection of all such three-dimensional pictures moving in time would give a four-dimensional portrayal of the events of the lifetime.

"In a similar way," the mathematician explained, "A segment—a line—moving in a direction perpendicular to itself would describe a square region, and a square moving in a direction perpendicular to itself would generate a three-dimensional cube. We could think of the cube as the lifetime of a square, with any individual square slice representing first one event in that lifetime." The pictures on the television screen show how the slices fill out a perspective view of a cube.

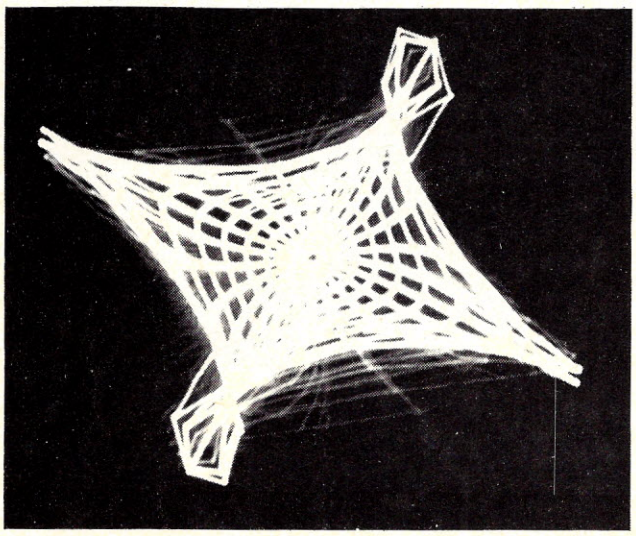
The next step moves into the fourth dimension.

"In three-dimensional space, we can't actually move the cube in a direction perpendicular to itself to generate a four-dimensional cube, or 'hypercube,' but we can see what the images of such a fourth dimensional would be as it passes through our three-dimensional space. At the turn of three control dials, the picture of a cube begins to move away from itself, tracing out a network that unfolds and collapses back in on itself. "We are actually watching the shadows of a four dimensional cube projecting down to our space."

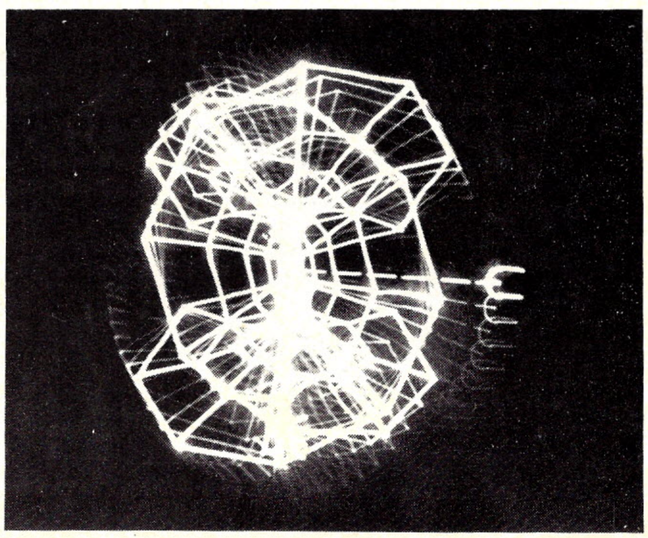
The hypercube is one of the least complicated of the objects that Prof. Banchoff and Prof. Strauss are studying. Some are difficult enough to require the use of a stereoscope so that the views of these projections and slices appear truly three-dimensional. Many of these ideas have potential usefulness in physics or economics, wherever quantitives of data occur which cannot be handled easily by ordinary three-dimensional representations. In many ways, the project is just beginning.

"One of our recent techniques involves describing an object by its slices," explains Prof. Banchoff. "A knife moving across a square produces a collection of parallel segments. A cutting plane moving across a cube and parallel to one face will produce a collection of squares. You might say that the cube is described as a stack of square slices, as a square is a stack of segments. In our work, we study the individual three-dimensional slices that will stack up in some direction perpendicular to all of our space to form a four-dimensional object." ■

*166 E. Main St., Norton, MA 02766



Algebraic curve in fourth dimension



Projective plane (surface in four-space)

Computer Correction of Optical Illusions

David A. Smith
Department of Mathematics
Duke University
Durham, NC 27706
and
Case Western Reserve University

A computer-driven plotter is a very handy tool for drawing optical illusions, especially those that depend on distortion patterns consisting of many curves or that involve complicated curves that must be drawn accurately. In a recent study (3) we have shown that the same computer plotter, given a little help from mathematics, can also "correct" illusions, in the sense of causing to appear what was really there in the first place, but didn't seem to be. As we shall see, a corrected illusion is really another type of illusion. Each of the accompanying illustrations shows an illusion in part (a) and a corresponding corrected illusion in part (b). We will discuss them individually below, but in part (a) all the lines are straight and the curves are drawn without fudging, whereas in part (b) some very specific fudging has been provided by the computer program.

It should be noted that this article is not about mathematical recreations. Understanding how and why our eyes play tricks on us is an important step in understanding the structure and function of the human visual system and its various parts. This is an area of active research by physiologists and psychologists of perception, and many important questions remain unanswered. An important preliminary step is to describe accurately what we perceive and how it differs from what is really there. One measure of accuracy of a descriptive model for perception of illusions is its ability to reverse the perceived effect and produce a corrected illusion that appears to show what did not appear when the figure was drawn "correctly."

The underlying principle on which the descriptive mathematical model is based was first hypothesized by F. Brentano about 80 years ago: *The human visual system tends to overestimate acute angles and underestimate obtuse angles.* This is certainly not an immutable law of nature, but there is a great deal of evidence to support its validity, at least among adults in civilized societies, living in a largely "rectangular" environment. The Brentano Hypothesis may be formulated precisely and applied to plane optical illusions by means of two-dimensional vector calculus. This leads to algebraic or differential equations whose solutions describe the correction curves. Each such equation contains a single numerical parameter representing "how much" misperception is taking place at each angle in the illusion, which we call the *strength* of the illusion. Some of the factors affecting the strength parameter, within the drawing, within the visual system, and related to the manner in which the drawing is viewed, are known, but its precise relationship to neurophysiology has not yet been determined. Values of the strength parameter for each of the corrected illusions shown here were determined empirically, by drawing the corrections for several closely-spaced values and picking the one that looked best. (A much more intricate mathematical model that supports the Brentano Hypothesis and is based directly on the physiology of the eye has been given by E.H. Walker(4).)

Our technical paper (3) was illustrated with classical illusions, much studied in the psychological literature, and associated with the names of Poggendorff, Zollner, Hering, Orbison, Ponzo, and Muller-Lyer. The illusions presented here are unconventional, but each has a purpose in supporting the Brentano model. The model is based on work of W.C. Hoffman (1), but with some specific differences of detail that contradict Hoffman's theory of perception. In particular, Hoffman's theory led him to predict the possibility of illusions with hyperbolic or spiral distortion patterns (see Figure 1), but not with sinusoidal distortion patterns or distorted curves (see Figures 2, 3, and 4). Indeed, Hoffman has since stated (2): "(A) spiral, representing a combination of size and rotation constancies, may be involved in such a visual illusion, but a sinusoid, for example, cannot."

Figure 1 is a variant of the classical Orbison illusion, in which a square is drawn in a pattern of concentric circles. In that case, the sides of the square appear to be bowed inward, whereas they appear to be bowed outward when the distortion pattern consists of rectangular hyperbolas. In Figure 1 (b), the sides of the "square" appear to be straight, but they are actually bowed inward, as you can verify by holding the page at eye level and sighting along one of the lines.

Figure 2 is a variant of the familiar Poggendorff illusion, in which a diagonal line is interrupted by two parallel lines. The diagonal appears to be parts of two different lines. Here the interrupted curve is a sine wave, specifically, $y = \sin 2x$. Because of the angular intersections with the parallel lines, the two portions of the curve do not appear to be smoothly connected "behind" the vertical strip (unless you look along the curves at eye level). The possibility of this illusion was specifically denied by Hoffman in (2). Figure 2 (b) was drawn with $y = \sin 2x$ up to the peak on the left, then with $y = \sin (2.1745 x - 0.13705)$ from the peak to the left-hand vertical, with the right hand curve being a mirror image. (Because of the nonlinear equations involved, the computer was also needed to find the right numbers.)

Figures 3 and 4 use Orbison-type distortion patterns composed of closely-spaced sine waves. In Figure 3 the curves have equal period and varying amplitude, while in Figure 4 the curves differ by vertical displacement. The

correction curves are produced in the same manner as those for Figure 1, by numerical solution of a differential equation.

Figure 5, which we call the "oriental lantern," is a variant of the Muller-Lyer "arrowhead" illusion. It illustrates how misperception of angles can create an illusion of *length,* as opposed to shape or angle. In Figure 5 (a), all of the horizontal lines have exactly the same length, but the five "outer" lines appear longer than the four "inner" ones. In Figure 5 (b), while the horizontals *appear* to be of the same length, the "inner" ones are somewhat longer, as you can verify by laying a straightedge on the figure. In this case, the correction is done by adjusting the slopes of the diagonal segments according to the Brentano Hypothesis, and then filling in the horizontals.

The drawings in Figure 1 were done on a Complot plotter at Duke University Computation Center, and the remaining figures were done on a Calcomp plotter at Chi Corporation, a subsidiary of Case Western Reserve University.

REFERENCES

1. W.C. Hoffman, "Visual illusions of angle as an application of Lie transformation groups," *SIAM Review* 13 (1971), 169-184.
2. W.C. Hoffman, "In defense of LTG/NP," to appear (presented to the American Mathematical Society, August 19, 1975).
3. D.A. Smith, "A descriptive model for perception of optical illusions," submitted to SIAM Review.
4. E.H. Walker, "A mathematical theory of optical illusions and figural aftereffects," *Perception and Psychophysics* 13 (1973), 467-486.

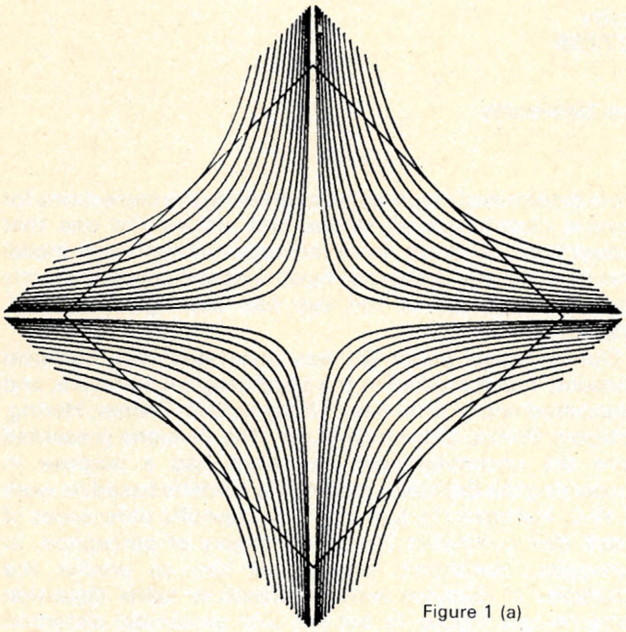


Figure 1 (a)

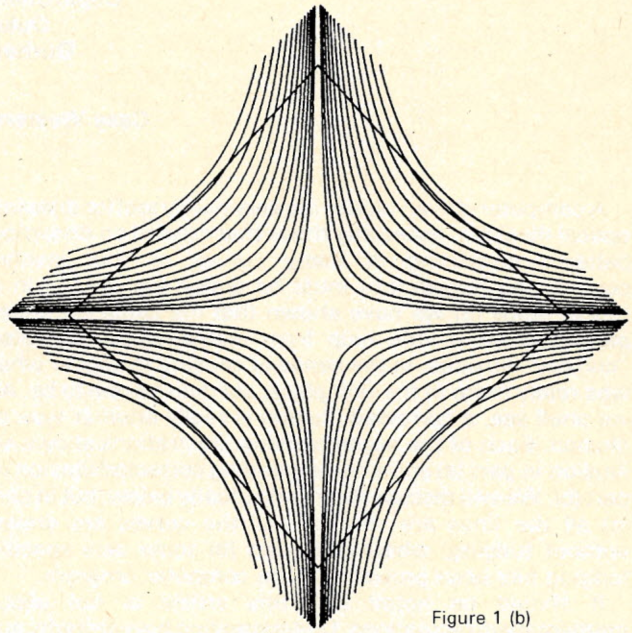


Figure 1 (b)

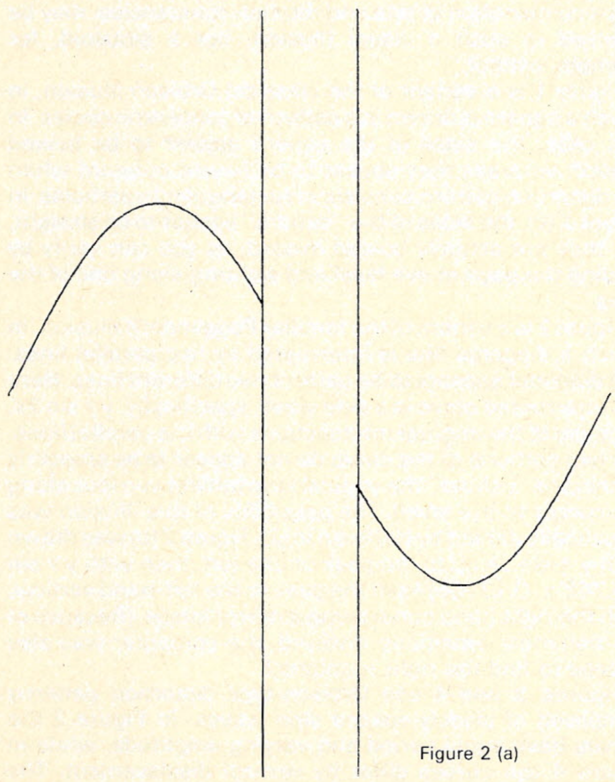


Figure 2 (a)

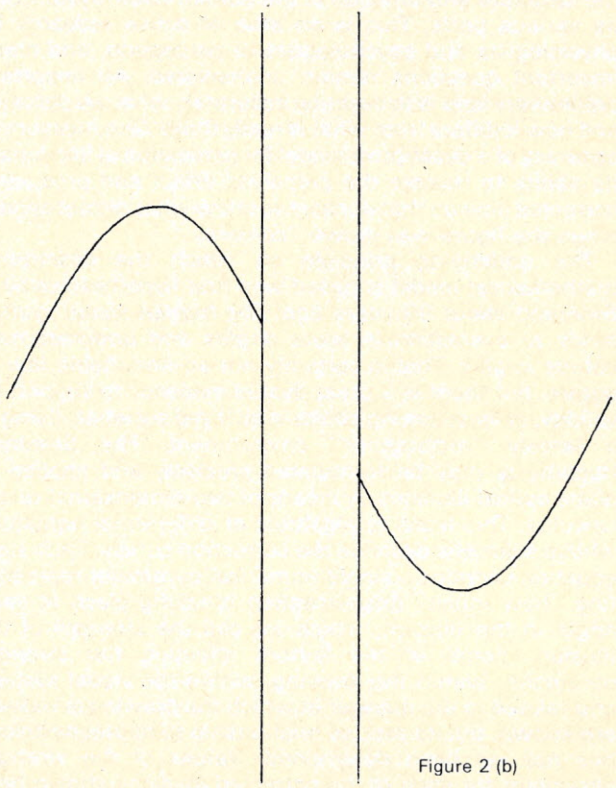


Figure 2 (b)

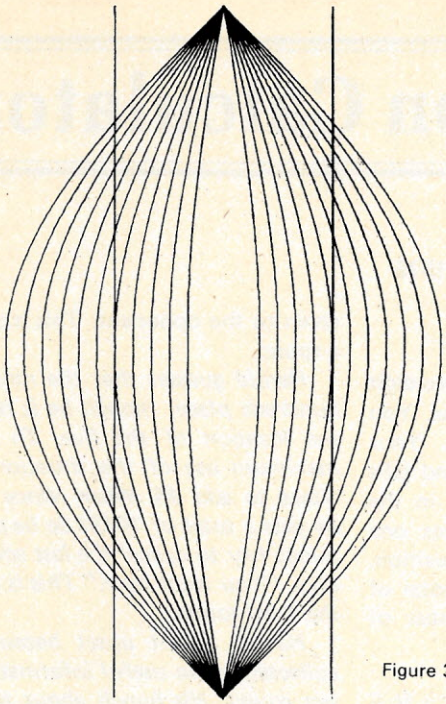


Figure 3 (a)

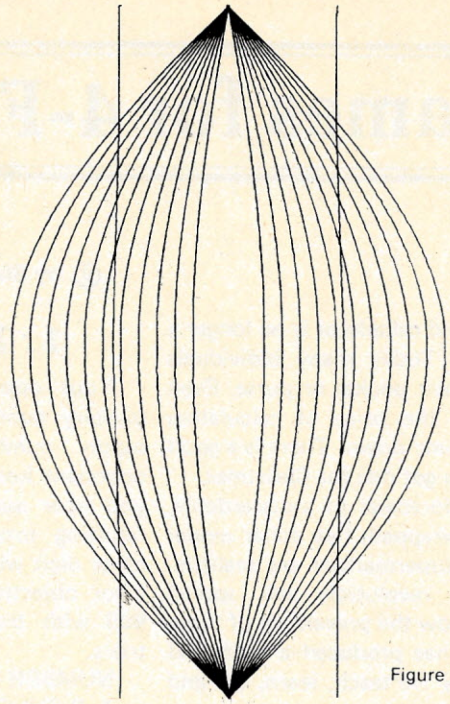


Figure 3 (b)

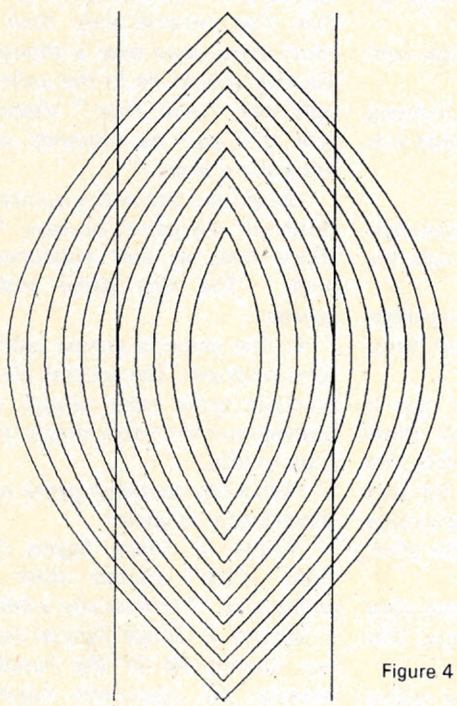


Figure 4 (a)

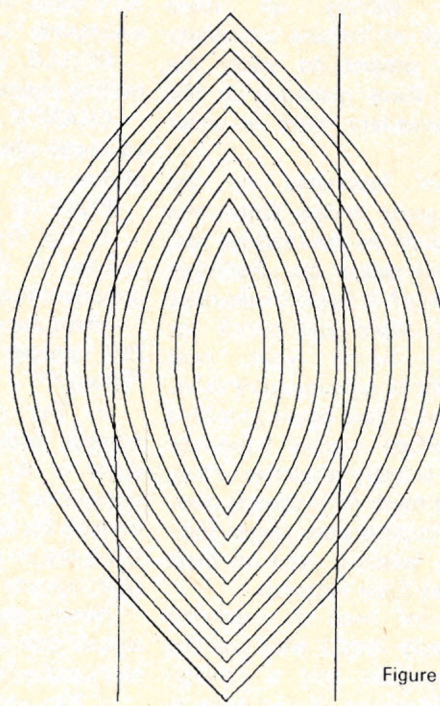


Figure 4 (b)

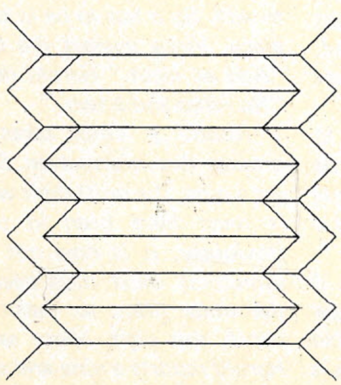


Figure 5 (a)

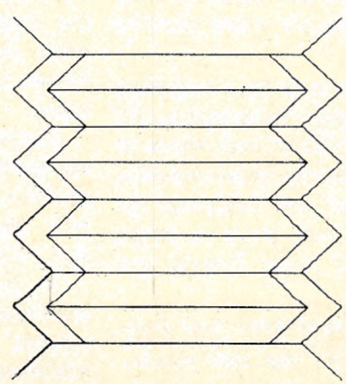


Figure 5 (b)

4 Games for 4-Function Calculators

© by Sivasailam Thiagarajan 1975

The hand-held calculator is no longer a status symbol. Today many individuals carry one in their pocket or purse. Even with inflation, the price of calculators continues to come down. There is a good chance that you got one for Christmas.

If industry forecasts of a discardable, plastic-bubble-wrapped, five dollar model to be sold in supermarkets are realistic, the calculator revolution will spread rapidly. Even now the prevalence of these handy devices has produced a profound shift in the way we teach, learn, use and perceive math.

While use of calculators in the classrooms is stirring up some controversy, back at Indiana University an NSF-funded project to study the implications of these machines in the teaching of elementary math is under way.

I have been fascinated by the calculator as a game device. Where else can you find such an inexpensive and portable electronic gadget that challenges your mental agility? A pocket calculator is handier than paper and pencil for playing games during your next transatlantic flight or cross-country drive.

My friends and I have come up with a dozen games which involve hand-held calculators. Our primary objective is fun; any insights into math or increased competency with calculators are purely incidental. I have selected four of these games which share these features:

1. You do not need an expensive calculator. A cheap model with just the four basic operations and a six-digit display is fine. However, if you have an SR-50 or HP-65 you can come up with variations which provide more exciting options.

2. You do not have to be a high-level mathematician to play these games. All the games permit children and adults to compete on satisfactorily equal grounds.

3. You don't need one calculator for each player. You can share a single instrument with all your friends and opponents.

Reprinted with permission of the author. These games originally appeared in the Jan. 1975 *Simulation/Gaming/News*, P.O. Box 3039, University Station, Moscow, ID 83843.

1 : Triple Nine

Your opponent enters a three-digit mystery number in the calculator. You supply numbers to be added. Your opponent keeps a running total and tells you how many nines there are in the resulting totals. He also identifies one other digit without revealing its position. Your objective is to reach the total of 999 with the least possible number of trials.

NUMBER OF PLAYERS: Two.

APPROXIMATE TIME REQUIREMENT: Three to seven minutes for a game. Fifteen to thirty minutes for a complete "match."

SKILLS INVOLVED: Addition and making logical inferences.

CHANCE LEVEL: Very little. Among advanced players the game becomes one of pure skill.

PLAY OF THE GAME:

1. The first player enters a three-digit number in the calculator and tells his opponent that he is ready.

Vince punches in the mystery number 297, presses the plus key and says, "Ready!"

2. The second player calls out a number which has one, two, or three digits in it. The first player adds this number to his original number. He then informs his opponent (a) how many nines there are in the total and (b) any other digit, but not its position.

Harold says, "Add 123." Vince does so and gets a total of 420. He says, "No nines and a four."

3. The process of the opponent calling out a number, the first player adding it to the total and giving information about nines and one other digit, is repeated. Players keep track of how many turns are taken.

Harold says, "Add 555," hoping to change the four to a nine regardless of its position. Vince adds the 555 and gets 975. He tells Harold, "One nine and a five." This is the end of round two.

4. During any round of the game, if the total goes over 999, the first player returns to the previous total. No additional information about the digits is given to the opponent. This is counted as a round.

Harold guesses that the nine is in the hundreds place, though he is not sure of the location of the five. To make the maximum use of the situation, he asks Vince to add 44. When Vince does this, he gets a total of 1019! So he presses the minus key to cancel the last addition and says, "You went over." This is the end of the third round.

Harold is not upset because he has collected some useful information in the last round. His hunch about the nine in the hundreds place is confirmed. He also figures out that the tens digit is greater than five because only then the total could have gone over a thousand. Also, the five should be in the units place. He calls out, "Add four." Vince's total is now 979 and he responds with "Two nines and a seven."

Harold has the entire number now. To finish off the game, he says, "Add 20." Vince does so and announces "Three nines." The first game ends in five rounds.

5. The game is played again with the roles reversed. The second player in the previous game now selects a mystery number and the other player tries to run it up to 999.

Here's the complete game when it was Vince's turn to guess:

Round 1. Vince begins by saying, "Add 123." Harold does this and announces, "No nines and a one."

Round 2. Vince figures out that the one cannot be in the hundreds place because his opponent began with a three-digit number and he gave him a 123 to add. So it has to be in the tens or units place. Vince guesses the latter and says, "Add an eight." Harold does so and says, "No nines and a two."

Round 3: Vince takes a moment to process this information. Since he did not get a nine, the units digit was not the one. It must have been in the tens place. Obviously, it is now not a one because something must have been carried from the units place to make it a two. To clinch this digit, Vince says, "Add 70." Harold reports a nine and a three.

Round 4. Where is this three? Since this is the fourth round, Vince doesn't think it is in the hundreds place. "So," he says, "add six." His guess was wrong: His opponent announces the same nine and a three.

Round 5. Apparently the three was—and still is—in the hundreds place. Vince says, "Add 600." As he had expected he gets two nines. He also gets a six.

Round 6. Vince has the total picture now. He says, "Add a three" and gets his triple nines.

6. Two games make a set. The player who gets three nines with the least number of rounds wins the set.

Since Vince needed six rounds and Harold only five, Harold wins.

VARIATIONS:

1. With younger players you can play the double nine game. With more advanced players you can try quadruple nines.
2. Try a triple seven game with the guessing players permitted to use either addition or subtraction.
3. Play the countdown game of reducing a three-digit mystery number to a zero through a series of subtractions.

"Make it 76!" says Thiagi as he hands over the calculator to Robbie Stolovitch, the youngest player in this group trying out the game PLUS OR MINUS. To compensate for the obvious disparity in ages, these are the restrictions imposed: From left to right, Harold Stolovitch can be required to change the display to any other two-digit number. Raja Thiagarajan furnishes only those numbers which require addition without carrying or subtraction without borrowing. Thiagi can get stuck with any two-digit number, while Robbie handles only simple additions involving one, two or three. (Photograph by Len Peak)

2: Who's Closer?

Each player in this game writes down a secret two-digit target number. Players take turns naming a single-digit number which is used by the next player to add, subtract, multiply or divide the number on display. Any player can freeze the display at any time. The player whose target number is the closest to the display wins the game.

NUMBER OF PLAYERS: Two to five.

APPROXIMATE TIME REQUIREMENT: Three to five minutes.

SKILLS INVOLVED: All four basic operations. There is also some scope for bluffing.

CHANCE FACTOR: Medium. Winning the game is based both on chance and skill.

PLAY OF THE GAME:

1. Each player writes down a secret two-digit number on a piece of paper and keeps it hidden from the others.

There are five players in this game. Stan's secret number is 57.

2. A random number is entered in the calculator by someone covering the display and each player punching in a single-digit number and the plus key.

The players punch in the following numbers, one at a time: 7, 9, 1, 3, and 5. This gives a total of 25 which is revealed to all players.

3. The first player calls out a single-digit number. The next player may perform any operation with this number and the one on display.

Toby calls out a three. Harold decides to use this number to multiply the 25 on display to get 75.

4. The game continues with players taking turns to call out the single-digit number and use it in some fashion. Any player can freeze the number on display at any time.

Harold calls out the number nine. Stan, the third player, subtracts it from the 75 and gets a 66. He calls out a seven and hands over the calculator to Vince. Vince subtracts this seven from 66 to get 59. Since this is very close to Stan's target number of 57, he calls out "Freeze."

5. All players compare their target numbers with the frozen display. The player whose target number is the closest to the frozen number wins the game.

The players reveal their target numbers: Toby, 12; Harold, 99; Stan, 57; Vince, 32; and Maggie, 60. Stan is close, but not close enough. Maggie with her 60 is the closest to the frozen 59 and wins the game. She did not open her mouth once during the game, but won anyhow!

6. The next round of the game begins where the previous one ends.

In this case, Maggie calls out the first number. This method ensures that no one is out of action for a long period of time.

VARIATIONS:

1. Try the game with three-digit targets and two-digit call-out numbers.
2. You can remove all restrictions on the target number and use exponentiation, extraction of the n-th root, and other such complex maneuvers.

3: Nim

Here's an exciting way to play this popular numbers game. For those who don't know how to play NIM, you take turns to add numbers five or less to a running total. The player who reaches the total of 25 wins the game.

Technically, NIM is a "trivial" game in which the first player always wins if he makes the right moves. This is just the game for you to beat the champ.

NUMBER OF PLAYERS: Two.

APPROXIMATE TIME REQUIREMENT: Three to five minutes.

SKILLS INVOLVED: Simple addition. Ability to induce a formula for consistently winning the game.

CHANCE FACTOR: No chance is involved if you know the system.

PLAY OF THE GAME:

1. The first player enters 1, 2, 3, 4, or 5 in the calculator.

Toby is the first player. She enters a one and presses the plus key.

2. Players take turns adding five or less to the number on display.

Stan adds a three to the one and gets a total of four. From this point on, the game proceeds as follows:

Toby	*Stan*
4 + 3 = 7	*7 + 2 = 9*
9 + 4 = 13	*13 + 5 = 18*
18 + 1 = 19	*. . .*

3. The game ends whenever the total reaches 25. The player who arrived at this total is the winner.

At this stage of the game, Stan is stuck. He cannot make the 19 become a 25 because he may not add a number greater than five. Whatever other number he adds, Toby can come up with the clincher. For example, if he adds one, she adds five; if he adds two, she adds four, and so on.

4. The second player in the previous game begins the next game.

Stan enters the first number to begin the next game. He chose the number five. Here's how the game goes:

Stan	*Toby*
5	*5 + 2 = 7*
7 + 5 = 12	*12 + 1 = 13*
13 + 3 = 16	*16 + 3 = 19*
. . .	

Stan is stuck again. Obviously reaching the number 19 is a part of the system!

VARIATIONS:

You can change both the total number to be reached and limit for the numbers added during each turn. For example, you can try to reach a total of 365 using any number less than 30.

WINNING STRATEGY:

If you play the game repeatedly, you will probably figure out how to win it every time. In case you are in a hurry to show off to your friends, here's the system:

Let's first take the game in which the target is 25 and each number added must be five or less. Here are the moves:

1. If you are the first player, begin with the number one.

2. Subtract whatever number your opponent adds from six. When it is your turn to add, use this difference. (If your opponent adds a 1, you add 5 (i.e., 6-1). If your opponent adds a 2, you add 4 (i.e., 6-2). And so on. . . .)

3. If your opponent begins the game, watch out for these intermediate totals: 7, 13 and 19. Try to arrive at them during any convenient round in the game. From then on, you can win the game by using the second strategy move above.

For more advanced players here's the formula for winning NIM with any other target total and added restrictions:

1. If **N** is the target total and **n** is the number you may not exceed during each addition, then find N/(n+1). Disregard the quotient; the remainder you get is the number to begin the game with.

For example, if you are trying to reach 99 by adding any number less than nine, N = 99, n = 9.

99/(9+1) = 9, remainder 9. You should begin the game with this remainder 9.

2. Subtract the number your opponent adds from (n+1). The difference you get is the number you should add.

If your opponent adds 3, you should add 7 (i.e., 10-7).

3. If you subtract (n+1) repeatedly from N, you get the intermediate totals to shoot for if your opponent begins the game.

The intermediate totals you should shoot for in this game are 89, 79, 69, . . . which are obtained by repeatedly subtracting 10 from the target total. Once you reach any of these numbers during play, apply the second strategy listed above.

4: Plus or Minus

Any number can play this fast-moving game of the "Buzz" variety. Players pass the calculator around, requesting the next player to change the number on display to something else through addition or subtraction. If the player makes an error or exceeds the time limit, he drops out of the game. The game continues until a sole survivor is identified.

NUMBER OF PLAYERS: Two to ten.

APPROXIMATE TIME REQUIREMENT: Depending upon the players' computational skills, the game may last anywhere from a few minutes to a few days. We once had a game going for three days, off and on.

SKILLS INVOLVED: Ability to add and subtract two-digit numbers rapidly.

CHANCE FACTOR: None.

PLAY OF THE GAME:

1. Players are seated around in a rough circle. The first player enters any two-digit number in the calculator and passes it to the player on his left. He then calls out another two-digit number.

Gavin is the first player. He enters 36, presses the plus key, gives the calculator to his neighbor Lillian and says, "Fifty one."

2. The player receiving the calculator changes the number on display to the requested number through a single addition or subtraction. He does this within a count of 10.

As soon as Lillian gets the calculator, Gavin starts counting slowly to 10. Lillian plugs in 15 and presses the plus key to get the requested number, 51.

3. The game continues with each player calling out a new two-digit number and passing the calculator to the neighbor.

Lillian calls out 27, passes the calculator to John and starts counting to 10. John performs the correct subtraction and passes the calculator to Ted with a request for 83.

4. If a player does not complete the task within the count of 10, or if he does not get the required number, he drops out of the game. Before he leaves, however, he passes the calculator to the next person with a new request.

It takes Ted a count of 14 before he manipulates the number into 83. So he is eliminated from the game. Before he leaves the circle, however, he calls out 16, gives the calculator to Len and starts his count. Len gets flustered and subtracts 57 and gets 26 instead of 16. He is eliminated also, but passes the calculator to Jeff and requests 62.

5. As the game continues, only those players who are not eliminated in the previous rounds participate in the subsequent rounds. The last remaining player is the winner.

VARIATIONS:

When you are playing with people of different levels you may place restrictions on the requested number as illustrated in the photograph. You can also let younger players use a two-stage operation. Thus a child can go from 86 to 59 by subtracting 30 and adding three. You can also provide the younger players with a longer time limit. ■

An Inexpensive Reading Machine For The Blind

John M. Brus

This September a student at the Watertown, Mass. Perkins School for the Blind will put a textbook face down on a glass plate and a machine will 'read' the words to him.

The machine is a minicomputer based system that converts printed text—of any typeface—into spoken language at rates up to 200 words per minute. And it is the product of years of effort by Raymond Kurzweil, 28, and his small Cambridge, Mass. computer firm.

Blindness, affecting about 1.7 million Americans, frequently stems from disorders which can leave an individual multiply disabled. Consequently, a large percentage of the blind can never learn the Braille language. Add the fact that at most only three per cent of the books published in the United States are ever transformed into recorded form and the need for Kurzweil's machine becomes clear.

The U.S. Government agrees and is now financially backing the completion of the machine's development through the Veteran's Administration and the Bureau of Education for the Handicapped. The National Federation for the Blind is also helping financially and the machine will undergo field testing for the next 18 months before being placed on the market.

Kurzweil has combined engineering and programming ingenuity with the shrinking prices of semiconductor components to produce a product which he predicts will only cost $5,000 in five years.

How Does It Work?

Currently using a Data General Nova 2 minicomputer, the machine employs a series of programs using Assembler language to identify the individual letters and transform them into recognizable speech. "What we've done that's new is produce multiple typeface character recognition in an inexpensive minicomputer system," Kurzweil explains.

The basic sequence of operations begins when the camera—using an integrated scanning array—scans the letters in each word and feeds the data directly into the processor. A 'shape analysis' program identifies the letter and converts it into an eight-bit ASCII code. If necessary, other programs using contexual and other clues assist in the identification. Sequential algorithms then take the ASCII coded information and, using phonetic rules, transform the letters into phoneme signals. A phoneme is the sound associated with each letter. These signals in turn drive the Votrax speech synthesizer circuits.

Hundreds of typefaces are available to printers today and all vary according to line thickness, size and serifs. To deal with this variety, Kurzweil devised what he calls a 'Topological Analysis Program.' It recognizes that each letter has certain invariant relationships of shape and the program matches each letter to certain assigned 'property sets.'

For instance, Kurzweil defined an upper case "A" as a loop, a North-Central to South-West line segment, a North-Central to South-East line segment and one South concavity. Thus, the property sets are defined totally by lines, loops, concavities, vertices and their relationships. And on the average, five binary property sets define each letter.

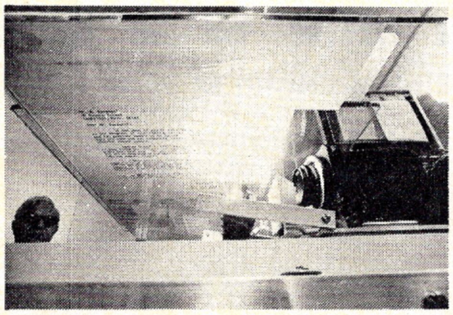
Sometimes this initial program cannot make a final identification. Then the output is an ambiguity code indicating several possible identifications. For example, a vertical bar might be an "I" or an "i" or the number "1." A back-up program makes final judgment using clues such as letter size, positional factors, presence of dots above or below and contextual clues.

Kurzweil gives an example of a contexual clue this way. "A vertical bar with a space to left and a consonant to the right is probably a capital "I" rather than a small "l" because words starting with "I" do not have second letter consonants."

If sloppy printing joins two letters, another program can identify the likely split points for the touching letters by analyzing the vertical-horizontal ratios of the image.

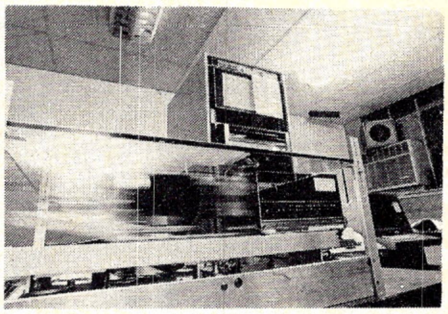
Word Pronunciation

After identifying and storing each letter, determining word pronunciation was the next programming problem. And since most of the processor was occupied by the character recognition work, there wasn't enough memory remaining for a standard 100,000 word English dictionary.



The basic mode of operation of the Kurzweil Reading Machine is automatic. The user simply places printed reading material face down on the glass plate of the desk top reading unit and presses a button to begin the scanning.

James Gashel, chief of the Washington office of the National Federation for the Blind, operates the simple controls of the Kurzweil Reading Machine, the world's first multi-font, full word speech reading machine for the blind.



In this time-lapse shot, the "scanner" converts a typewritten letter into digital signals for analysis by the computer in the electronic control unit (right), which contains the character recognition subsystem, the speech subsystem, and the synthesizer.

Raymond Kurzweil, president of Kurzweil Computer Products and inventor of the Kurzweil Reading Machine, points to the machine's "scanner," a small electronic camera moving on linear bearings and transmitting light and dark images of a printed page to the computer in the control unit.

Kurzweil's solution was programming a set of 1,000 phonetic rules for the English language and an additional 2,000 entry dictionary for exceptions. Both rules and exceptions were "practical for a minicomputer based system since the rules only required about 4,000 16-bit words of storage," according to Kurzweil. And to increase the effectiveness of the rules and exception dictionary, he added another program that can strip off prefixes and suffixes.

A final speech system provides a 'stress contour' across each sentence. He says this reduces the monotony of the speech and is intended to reduce listener fatigue. The synthesizer circuits give the machine's voice a thick Swedish accent that takes getting used to, but Kurzweil says most people can readily understand the voice with a couple hours practice.

However, the system cannot differentiate the pronunciation of about 20 'ambiguous' words. For example, 'lead' in 'lead magnet' pronounces differently from 'lead' in 'You lead, I'll follow.' But Kurzweil cites cost as the rationale for dismissing the problem, not technical difficulty.

Once all the bugs inherent in such a complex system are resolved, he plans to market the machine initially to institutions and then to leasing companies catering to individuals. ■

RAILROAD TALK

Tom Korb

Starting this year, all the nation's railroad cars, from engine to caboose, will have one thing in common. All will have a 12x26 inch color coded information sign attached to their sides.

Chances are, you've probably seen these labels on boxcar sides already, and if you are like most people, thought nothing of them. If however, you were inquisitive, and tried to "break the code", you'd find it next to impossible without a book, as, unlike languages, where certain vowels, words, and phrases are used more often than others, no two cars can have the same identification or configuration. Because of this seemingly disarray of patterns, people tend to think of them as some sort of decoration.

Nothing, of course, could be further from the truth. Every color, every combination of colors, and every position has a meaning. Every one of the thirteen printed modules in a placement is significant to a photometric scanner.

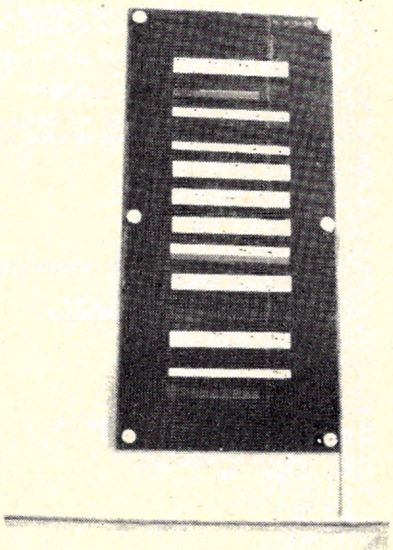
Recorded in the placements are four pieces of vital information; the first being, type of equipment. This can be anything from a locomotive on down, including the piggyback trailers and containers. Next comes the equipment owner, a serial number, and a validation check to verify that all the information is correct.

Because the system is based on the principle of geometric progression, (that is 1-2-4-8-16-32, instead of 1,2,3,4,5,) it is virtually foolproof. In the split second that it takes a railcar to pass by a scanner, a computer translates the colored modules into numbers. The numbers are then multiplied by varying degrees up to the ninth power, added together, and then divided, to get a verification. If any of the modules are ripped off or transposed, the computation will be different than the validation check, and the computer will notify the railroad immediately of a discrepancy.

The amazing part of this system, called AUTOMATIC CAR IDENTIFICATION, (ACI) is the fact that no train will have to be sidetracked to have a man do a car-by-car physical check on them. This will all be done automatically by scanners at checkpoints across the country, and the results put into a central memory bank, where any railroad can get instant information on their cars.

To the rail companies, this means huge savings of both time and money. There will be no more "lost" rolling stock. All equipment will be accounted for, and shipments can be traced on a day by day movement basis. Rush freight can truly be rushed now, as the precise location of cars will be known instantly, just by dialing in the central computer.

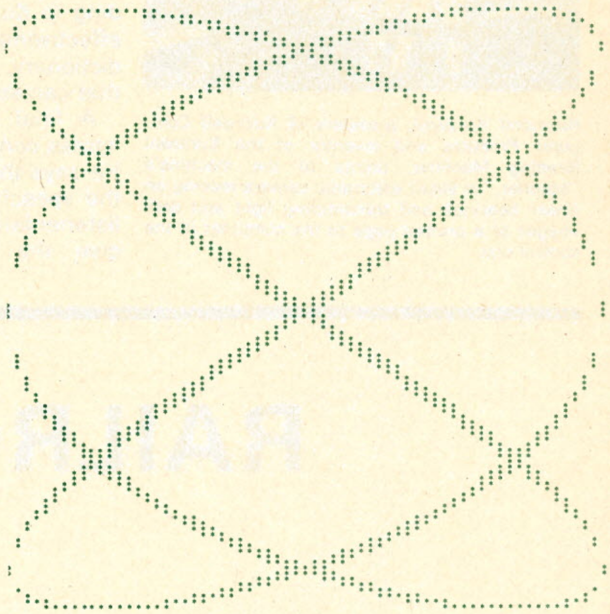
To you and me, this will mean faster delivery of goods. Ordered merchandise will get to us faster, and the fresh fruit and vegetables at the store will probably be just a little bit fresher. New jobs will be opening up as the system is expanded. Computer technicians and repair men will be needed as equipment is added and replaced. As always, there will be a need for railroad engineers; but these engineers will be working with slide rules instead of throttles, like Casey Jones did back in the all but forgotten era of steam.



CAR SIGN

This placement is saying to a scanner computer, "I am a caboose belonging to the Chicago and North Western. This is my serial number, and validation check."

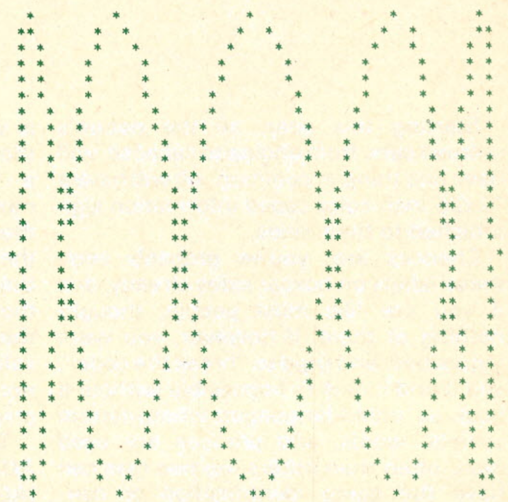
LISSAJOUS



This program prints Lissajous patterns. You enter relative x and y frequencies, and phase. Two versions of the program are listed here—one in MITS Extended BASIC, and one in IBM 370 BASIC for more primitive machines. The IBM 370 version permits you to specify the width of the lines in the pattern. Note that the program in MITS BASIC uses the FIX function. If your BASIC doesn't have this, try Y1=SGN(Y1)*INT(ABS(Y1)+.5), etc.

The Lissajous pattern plotter was written by Larry Ruane and modified by several other people.

```
10 DIM Y(10)
100 REM.  STEP-WISE LISSAJOUS
110 PRINT:P=3.14159
120 PRINT "RELATIVE FREQ. FOR X";:INPUT F1:IF INT(F1)<F1 THEN 120
122 IF F1<1 THEN 120
125 F=F1:F1=2*P*F1
130 PRINT "RELATIVE FREQ. FOR Y";:INPUT F2: IF INT(F2)<F2 THEN 130
132 IF F2<1 THEN 130
135 PRINT "Y PHASE, MULTIPLE OF PI";:INPUT P2:P2=P*P2
140 F2=2*P*F2
150 FOR X1=-18 TO 18
160 X=X1/18:GOSUB 1970:T1=X:T2=P-X
162 FOR I=0 TO F-1
165 T3=(T1+2*I*P)/F1:T4=(T2+2*I*P)/F1
170 Y1=30*SIN(F2*T3+P2):Y2=30*SIN(F2*T4+P2)
180 Y1=FIX(Y1+SGN(Y1)/2):Y2=FIX(Y2+SGN(Y2)/2)
190 Y(2*I)=Y1:Y(2*I+1)=Y2
200 NEXT I
210 FOR J=1 TO 2*F-1:I=J-1:T=Y(J)
220 IF T>=Y(I) THEN 240
230 Y(I+1)=Y(I):I=I-1:IF I>=0 THEN 220
240 Y(I+1)=T:NEXT J
250 FOR I=0 TO 2*F-1
260 IF I=0 THEN 280
270 IF Y(I)=Y(I-1) THEN 290
280 PRINT TAB(36+Y(I));"*";
290 NEXT I
300 PRINT
310 NEXT X1
1890 STOP
1960 REM:---------------
1970 IF ABS(X)<.1 THEN 2020
1980 X=X/(SQR(1+X)+SQR(1-X))
1990 GOSUB 1970
2000 X=2*X
2010 RETURN
2020 X=X+X↑3/6+.075*X↑5+X↑7/22.4
2030 RETURN
2040 END
```



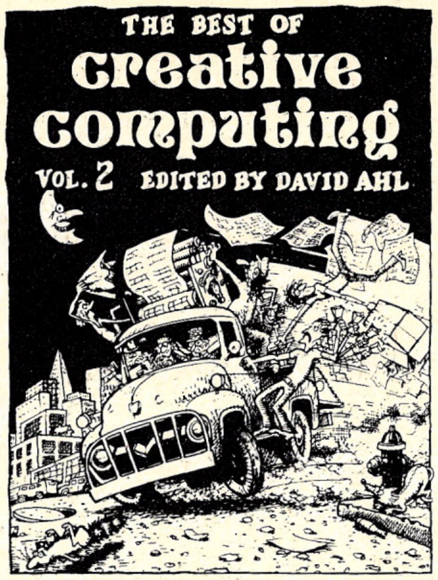
```
00010 REM - LISSAJOUS PATTERN PLOTTER  BY L. RUANE, ST. VIATOR H.S.
00020 REM - IBM 370 - HARPER COLLEGE, PALATINE, ILLINOIS
00030 DIM M(120)
00040 PRINT "ENTER FREQUENCY RATIO - FOR BEST RESULTS, ENTER SMALLEST NUMBER FIRST"
00050 INPUT A,B
00060 PRINT "ENTER PHASE - DEGREES"
00070 INPUT P
00080 PRINT "ENTER TOLERANCE - CONTROLS THE THICKNESS OF THE CURVE"
00090 PRINT "A GOOD VALUE IS BETWEEN 0 AND .2"
00100 INPUT T
00110 GO TO 370
00120 FOR Y=-32 TO 32
00130 F=ASN(-Y/32)
00140 H,K=0
00150 K=K+1
00160 FOR X=0 TO 1
00170 N=(X*(&PI-2*F)+F+2*&PI*K)*A/B+RAD(P)
00180 FOR C=-.5 TO .5
00190 FOR J=0 TO T*A/B STEP 3/80
00200 R=INT(SIN(N+J*C)*160/3+200/3)
00210 M(R)=1
00220 IF R<H THEN 240
00230 H=R
00240 NEXT J
00250 NEXT C
00260 NEXT X
00270 IF K*A/B>INT(K*A/B) THEN 150
00280 FOR X=1 TO H
00290 IF M(X)=1 THEN 320
00300 PRINT " ";
00310 GO TO 340
00320 PRINT "*";
00330 M(X)=0
00340 NEXT X
00350 PRINT
00360 NEXT Y
00370 FOR J=1 TO 5
00380 PRINT
00390 NEXT J
00400 IF Y=0 THEN 120
00410 END
```

The diversity in *The Best of Creative Computing — Volume 1* can only be described as staggering. The book contains 328 pages of articles and fiction about computers, games that you can play with computers and calculators, hilarious cartoons, vivid graphics and comprehensive book reviews.

Authors range from Isaac Asimov to Sen. John Tunney of California; from Marian Goldeen, an eighth-grader in Palo Alto to Erik McWilliams of the National Science Foundation; and from Dr. Sema Marks of CUNY to Peter Payack, a small press poet. In all, over 170 authors are represented in over 200 individual articles, learning activities, games, reviews and stories.

This 328-page book has 108 pages of articles on computers in education, CAI, programming, and the computer impact on society; 10 pages of fiction and poetry including a fascinating story by Isaac Asimov about all the computers on earth linking up after a nuclear war to support the few remaining survivors; 15 pages of "Foolishness" including a cute cartoon piece - called"Why We're Losing Our War Against Computers"; 26 pages on "People, Places, and Things" including the popular feature "The Compleat Computer Catalogue" which gives capsule reviews and lists sources for all kinds of computer-related goodies; 79 pages of learning activities, problems and puzzles; 29 pages continuing 18 computer games including a fantastic extended version of the single most popular computer game — Super Star Trek; and 32 pages of in-depth book and game reviews including Steve Gray's definitive review of 34 books on the Basic language.

The Best of Creative Computing - Volume 1 is available by mail for $8.95 plus 75¢ postage from Creative Computing Press, Attn: **Alyce** P.O. Box 789-M, Morristown, N.J. 07960.



This fascinating 336-page book contains the best of the articles, fiction, foolishness, puzzles, programs, games, and reviews from Volume 2 issues of *Creative Computing* magazine. The contents are enormously diverse with something for everyone. Fifteen new computer games are described with complete listings and sample runs for each; 67 pages are devoted to puzzles, problems, programs, and things to actually do. Frederik Pohl drops in for a visit along with 10 other super storytellers. And much more! The staggering diversity of the book can really only be grasped by examining the contents, or better yet, the book itself.

Price is $8.95 plus $0.75 shipping and handling in the USA ($9.70 total); outside USA, add $1.00 ($10.70 total). Individual orders must be prepaid. Creative Computing Press, Attn: Alyce P.O. Box 789-M, Morristown, NJ 07960.

THE BEST OF BYTE — VOL. 1

The Best of Byte - Volume 1 is a 384-page blockbuster of a book which contains the majority of material from the first 12 issues of *Byte* magazine. 146 pages are devoted to "Hardware" and are cram full of how-to articles on everything from TV displays to joysticks to cassette interfaces. The section on computer kits describes building 7 major kits. But hardware without software might as well be a boat anchor, so there are 125 pages of "Software and Applications" ranging from on-line debuggers to games to a complete small business accounting system. A section on "Theory" examines the how and why behind the circuits and programs, and a final section "Opinion" looks at where this explosive new hobby is heading.

The Best of Byte - Volume 1 is edited by Carl Helmers and David Ahl and published by Creative Computing Press. Price in the US is $11.95 plus $1.00 shipping and handling ($12.95 total); foreign orders add $1.00 ($13.95 total). Orders from individuals must be prepaid. Creative Computing Press, Attn: **Alyce** , P.O. Box 789-M, Morristown, NJ 07960. Allow 8 weeks for delivery.

101 BASIC Computer Games is the most popular book of computer games in the world. Every program in the book has been thoroughly tested and appears with a complete listing, sample run, and descriptive write-up. All you need add is a BASIC-speaking computer and you're set to go.

101 BASIC Computer Games. Edited by David H. Ahl. 248 pages. 8½x11 paperbound. $7.50 plus 75¢ postage and handling ($8.25 total) from Creative Computing, P.O. Box 789-M, Morristown, NJ 07960.

When does white equal blue?

The cover of this issue of **Creative Computing** *was created by Eric Morey using a process that breaks down a photographic image into tones of grey and "assigns" a color to each grey tone. An analog computer is used in their color-assignment process, which gives the user completely continuous control over grey-tone thresholds and width of color bands, as well as the actual colors. The results are dramatic and compelling.*

I spent several hours with Eric Morey, president of Metacolor, seeing a demonstration of the Metacolor process. We produced some 20 possible cover images and chatted about the process. Some of this interchange is printed below.

[I might speculate that this is the type of thing that could be done with a small digital computer, although Eric's system certainly offers extraordinary flexibility that would be difficult to duplicate. However, as readers get into this, be sure to keep us posted on your progress.] — DHA

Ahl: I'm talking with Eric Morey, of Metacolor. Let me see if I can describe this device. It seems to have around fifty different knobs on the front of a panel with some plugboards above the knobs, and basically it doesn't look anything like a computer at all. Next to it is another wooden box with a camera mounted in it and some sort of CRT screen in the back of the camera. Obviously the camera takes pictures of images on the CRT screen. Now tell me what it really is.

Morey: On the left is the copy stand. We have a light-box here and overhead lights so that we can feed in any kind of image from a 35-mm color slide or small or large negative, or any kind of transparency or any kind of an opaque image like a print, say a black-and-white print or a color print. Or a graphic rendering or typographic layout like a simple title layout, just black words on a white background. Almost any kind of image will do but we do have to have an original input. We don't digitally generate images in this system. We start with an image. We feed that image into the system with this video camera. It's a high-resolution black and white video camera. All it reads is tones.

Ahl: Now is the image digitized in there or is that a standard video camera?

Morey: It's a standard video camera. But there isn't anyplace in this process where the image is digitized. It remains as a video or analog signal throughout the process. The system goes into the analog computer, which analyzes the image in terms of its tones so, as the video signal comes through, it reads each point, each "pixel" in terms of its —

Ahl: What's a "pixel"?

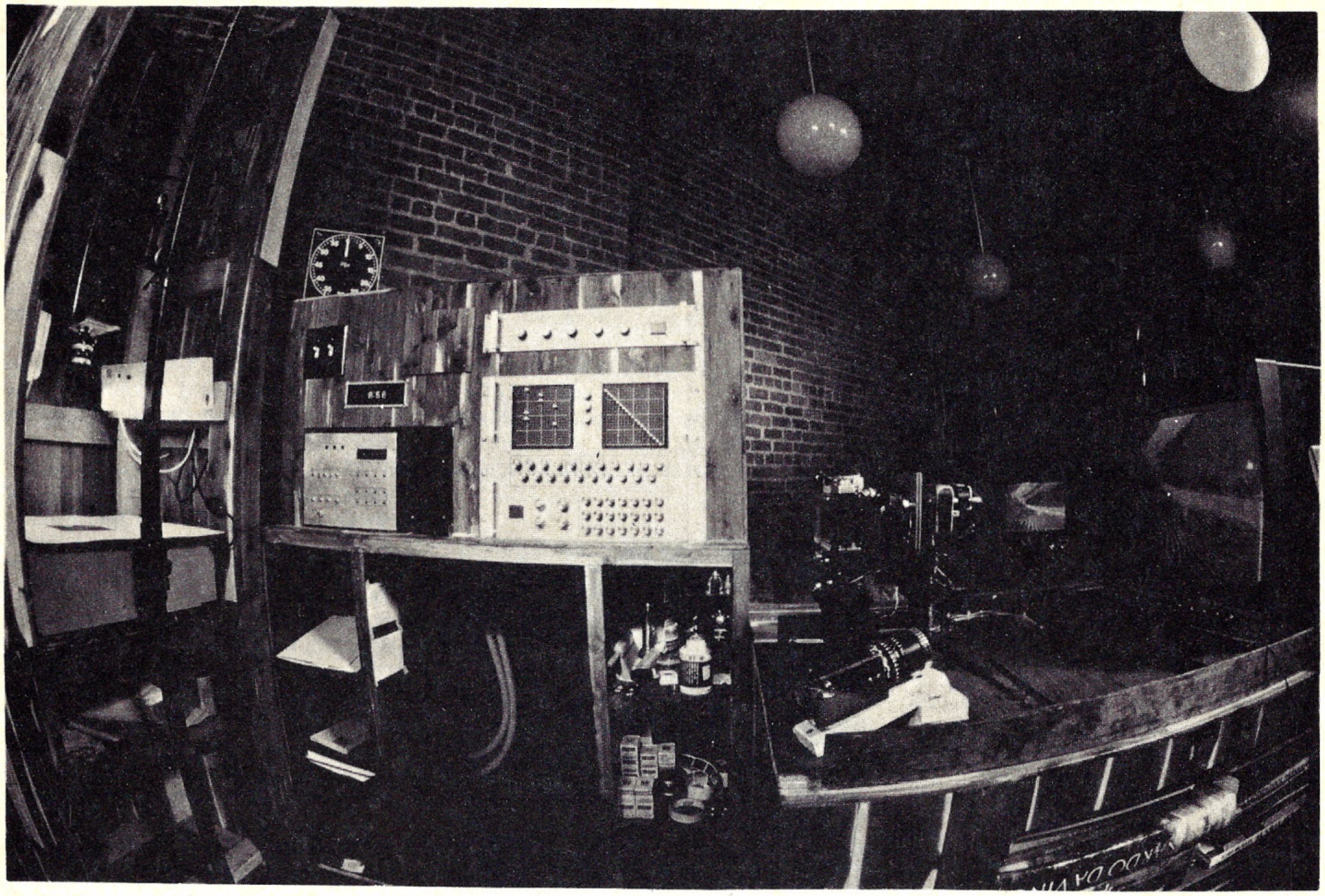
Morey: It's a point on a CRT. You have a 525-line scan system and 480 horizontal points on each line, so you have approximately 250,000 pixels or points on the image and it reads each point and analyzes it in terms of its grey scale, and we set up the grey scale to have any number of segments. We can say to the machine, in effect, "We want you to recognize only two segments of the grey scale. We want you to divide the grey scale into two segments or sections, and everything from white to middle grey will go in one category."

Ahl: In that case, you'd get an image like a photograph with only black or white.

Morey: Right, except that we could use two colors instead of black and white. We could use orange and purple, or green and brown. Any two colors. Or we can take three colors, break the grey scale down into three segments and put three colors into those segments. We won't have to break the grey scale down into even segments either.

We can take the first half of the grey scale from white to middle grey as one section. Here's a grey scale here that's

Fisheye view of the main control console of the Metacolor system, used to generate and enhance commercial images such as photographs, illustrations, and typographic layouts. On the extreme left is the video camera by which images are fed into the system. In the center are the controls for the analog computer, the special effects generator, and the exposure sequencer. On the right can be seen the remote-controlled motorized Nikon where the images are captured from a high-resolution television monitor. In the background is the display monitor where the operator (and the client, if he is present in the studio) view the final image.



a standard color grey scale that happens to have ten divisions on the scale and these are even divisions. We grade grey scales like this, but we don't have to do it evenly. We don't have to do ten, we can do two to 30. If we're doing ten of them, we can take everything from white to middle grey as one segment and then make our other nine divisions over on the darker half of the grey scale. They don't have to be even divisions. It's a very arbitrary system. We set it up for each job we shoot, for each image we work with, so then when we break the image down into its component grey tones, we program colors into those.

Ahl: Now if you're starting with a color image, you're still shooting it in black and white.

Morey: Right. This is a black-and-white camera, so we're only working with the tones. We're not working with the colors, so we can work equally well with a color photograph or a color slide. It's only working with the tones, not the actual colors. In other words, a red and a green that are the same intensity or the same tone, as opposed to their color, would read exactly the same and would come out the same color in our final product.

But if the red was a little darker than the green, or a little lighter, then we could separate them and work with them individually. So, as we do the design work, our client is generally sitting here in the studio watching the color TV monitor and he's making suggestions, he's saying, "OK, let's start out with a color palette or orange, purple, orange, magenta." I say, "OK, so we're breaking the grey scale into five segments and those are the five colors we're going to use." Then we put his image into the TV camera and we put those colors in his image and it appears on the color monitor instantly. I'll show you an example.

Here's an original image. I'm going to zoom in on it. It's a picture of a woman, face shot, and it happens to be a half-tone from a magazine, so really it's not even a straight photograph, it's a half-tone reproduction from a magazine in black & white. What you see now on the screen is simply the straight video signal going through into the monitor, not manipulated in any way. Now this is a program here and see, the program is made up of bands of color. Now I've got only one band of grey in it. I can put in another pin here, and this is where these patch boards come in.

Ahl: And the cross-hatching there is what, by how many?

Morey: Well, let's see, this is 20 by 20.

Ahl: And what do they represent in each direction?

Morey: Well, basically it's almost like a digital function. This patch board represents a yes-or-no function in terms of the color. If I put a pin here in the red section it just gives me a red band in the program. It doesn't tell me anything more than that I've got a red in. The position of the red band is controlled by other auxiliary elements down here so I can control the left edge and the right edge of that band. I can control the brightness of that band with these other controls. If I put this pin in the red down here, I get a red band in a different position because these knobs happen to be in different positions.

Ahl: So the knobs are controlling essentially the horizontal position of the vertical bar.

Morey: Right, and these knobs control the brightness and the intensity of the color.

Ahl: Now can you overlap?

Morey: Overlap the bands? Right. I can overlap this red onto that green,

see, and get yellow.
Ahl: So it's the same as mixing those two color guns or whatever is forming those colors.
Morey: Right, it's like having an electronic paintbrush. You can mix your own colors, say you want to mix a brown. You take a dark red and a darker green to get brown. Turn up the red a little bit and get into more of a rust color, an orange color, yellow, gold and finally, with both at maximum, to a bright greenish-yellow.

The bands are simply a program. This is simply the way we program them. This is like an IBM card.
Ahl: That's saying these are the colors I want to deal with.
Morey: Right, this is the palette.
Ahl: What we've got is five colors here, right?
Morey: I now have five bands and I'm now recognizing actually seven distinctions in the grey scale because the black here is at each end of the program or actually representing part of the grey scale too.
Ahl: So that there's actually three primary colors, in a sense, and two overlaps and two ends. Now you're applying those colors to that photograph. And by twirling different knobs, changing the threshold of greyness that each color corresponds to?
Morey: Well, essentially that's what I'm doing with this knob here, once I've got a program set up.
Ahl: Now we're back at the bands again. These bands correspond to levels of greyness from black on the left to white on the right.
Morey: Totally arbitrary. We can create as many as 30 of these bands and if we're doing 30 bands, we're dividing the image into 30 levels of grey which gets real messy and we never really do that. We work usually with four, five or maybe six levels of color.
Ahl: Now by selecting most of your image to fall into the red end of the spectrum, reds and oranges perhaps, you could achieve a kind of warm effect, so you could basically put more reds and oranges and really downplay the other colors.
Morey: Exactly. You could just say, "I'll pull the blue out of it and I could eliminate the green." There, I've got red and yellow and black, or if I want to fill that area there in with red, see, we can have only two colors.
Ahl: That's wild.
Morey: If I want to play with that green a little bit now, I can do some sort of very linear things here. What I'm doing right now for example, what you're seeing right now is an outline. I've got

one red band covering the whole grey scale and I've got a narrow green band overlapping the red, making a yellow in one section and this is the way it comes out. Now, see if I run this green band over here on the left side which is the shadow side, then I'm filling in those shadowy areas of the image. Those areas right around the nose and the eyebrow are the shadow areas of the image over the left side of the shadow area. If I move across to the right side, then I'm filling in just the highlighted area and I can go anywhere across. I can make these bands very narrow or very wide as I choose.
Ahl: That's really a dramatic effect. Just wild. With just two colors.
Morey: Let's see what happens. Add a third color here. One of the really amazing things about the system that really excites our customers is that not only do they have all these possibilities but they get to see the result immediately and they don't have to guess.

Good photolabs can do photo posturization, which is a term meaning exactly what we are doing, breaking the grey scale down into segments and then programming. Well, you don't have to use colors actually. You mentioned earlier a high-contrast photograph of just black and white. That's a posturization. That's the simplest kind of posturization. It's breaking that image down into just two tones and any grey that's darker than middle grey is going to fall into the black area and go black, and greys lighter than that will turn white.
Ahl: Who developed this machine?
Morey: This actually was built by Colorado Video, but the technology was developed by, I believe, Philco-Ford about 15 years ago. NASA wanted a way to determine from photographs how deep the craters were on the moon, and they realized that the angle of the sun and the darkness of the shadows were in direct correlation to the depth of the crater. But it's hard to look at a black-and-white photo and see exactly how deep it is, or see exactly how dark the grey tones are in it.

They realized that if they had a way of measuring the darkness of the shadows, they could measure the depth of the craters and of course, the deeper it goes, the darker it gets, so they developed this machine which would analyze these photographs. And take a picture of the crater itself and look at the grey tones and break them down into distinct segments and say ok, this is a 95% grey, this is an 80% grey, this is a 70% grey and so forth. And a 70% grey means it's 125 feet deep. One way they could do it was to have a digital printout of that picture, but they decided that as a visual aid they would develop this system which would turn those grey tones into colors so they could put pictures of the moon through this process and come out with a color picture where the colors would be directly correlated to the depth of the crater. So that green on the crater meant that it was 180 feet deep, purple meant that it was 200 feet deep and so forth. However, they haven't set it up. The system has been used to analyze earth resources from photographs.
Ahl: Yes, I've seen some of those satellite photos done this way. Did you buy this?
Morey: I bought this from Colorado Video. However, as it stands, it really is not suitable to do graphic arts work.

You have to have some peripherals to get images which are clear enough and with high enough resolution to apply it to graphic-arts applications.

All the people in NASA cared about was that they could see the color but a graphic designer wants to be able to see the details of the image and his client wants to be able to see the details of the image, so I developed these extra peripherals myself to go with it and really put together a whole system.

Ahl: Which are what peripherals?

Morey: Basically the reproduction unit is what you see here, and I won't bother about that too much because that's sort of our own development, but it's a way of taking that image and making a high-resolution color slide that's suitable for graphic-arts applications, magazine covers, slide shows.

Ahl: So what you've done is to use your talent in photography to figure out a way to photograph high-resolution images off your TV screen?

Morey: Right. Now to get that onto a piece of film. But if you just take a picture of that monitor, the resolution isn't high enough, the color TV part.

Ahl: Why is that?

Morey: Because a color TV monitor is made up, if you look at one closely enough, of dots. That's like a grain structure, in a sense, in a film and it's quite noticeable. And you take a slide of that and project the slide and you can see very clearly these dots and the scan lines.

Ahl: If I observe your set-up correctly, what you do is take a multiple exposure of a black-and-white TV image through three different filters representing the three color guns of a color TV set.

Morey: Right.

Ahl: That's very clever. Who are most of your customers?

Morey: Right now most of our customers are graphics people, doing brochure covers, annual report covers, magazine ads. Advertising and graphics people. Audio/visual producers who do slide shows use our output at a very reasonable cost and it opens up a whole area of possibilities and also for audiovisual producers we can do dissolve sequences where we take a slide like this, a slide like that, another one like that, etc., and if they flash this on the screen quickly with dissolving one image into the next, you've created movement, it's like doing animation.

We can record that movement on video tape or we can film off the monitor, we don't really have a way of filming with high-resolution systems, but in a film it's not as critical because it's moving and we can film in 16-mm off the monitor and do moving pictures with it. So we're now capable of putting out 35-mm slides or 16-mm movie film or video tape.

Ahl: What kind of images do you find work the best, ones without too much detail?

Morey: Exactly. The kinds of images that work the best are very graphic images where the impact is not dependent on a lot of fine detail, but has a good graphic formal quality to it.

Ahl: So in other words a portrait would do better than a group of people.

Morey: Exactly. Although you can do some nice things with a group of people, for graphic impact a portrait's better, like this one here, it's got a very graphic quality to it, the eyes, the nose, it's all describable through the shape, it's describing its content in terms of shape as opposed to little details, as for example a crowd of people. A crowd of people usually comes out to be an abstract mess. I've done a few shots with crowds, but we

also find that the kinds of images that we end up working with mostly are photographic images that for one reason or another are not suitable graphically for the application, let's say, for example, you're an electronics manufacturer, you want to do a magazine ad. You have a product which it does some really amazing things but . . .

Ahl: Everybody's printed-circuit board looks exactly like everybody else's.

Morey: They all look the same. They all look very dull. Well, that's not actually true. A printed-circuit board is pretty interesting looking. But still they all look the same. OK so you want to do something that makes yours stand out. You want to have an eye-grabbing image, so when somebody looks through that magazine, you want them

to stop for your ad. This is the kind of images we've done a lot of work with for the electronics companies. One of the reasons, we're near most of them and I find that electronics companies are pretty willing to experiment with graphic effects and also kind of . . .

Ahl: As opposed to what?

Morey: Well, food companies are out. Food companies don't bring us any business whatsoever because they don't have any real direct representational images.

Ahl: It's got to look like the food or the product that it is.

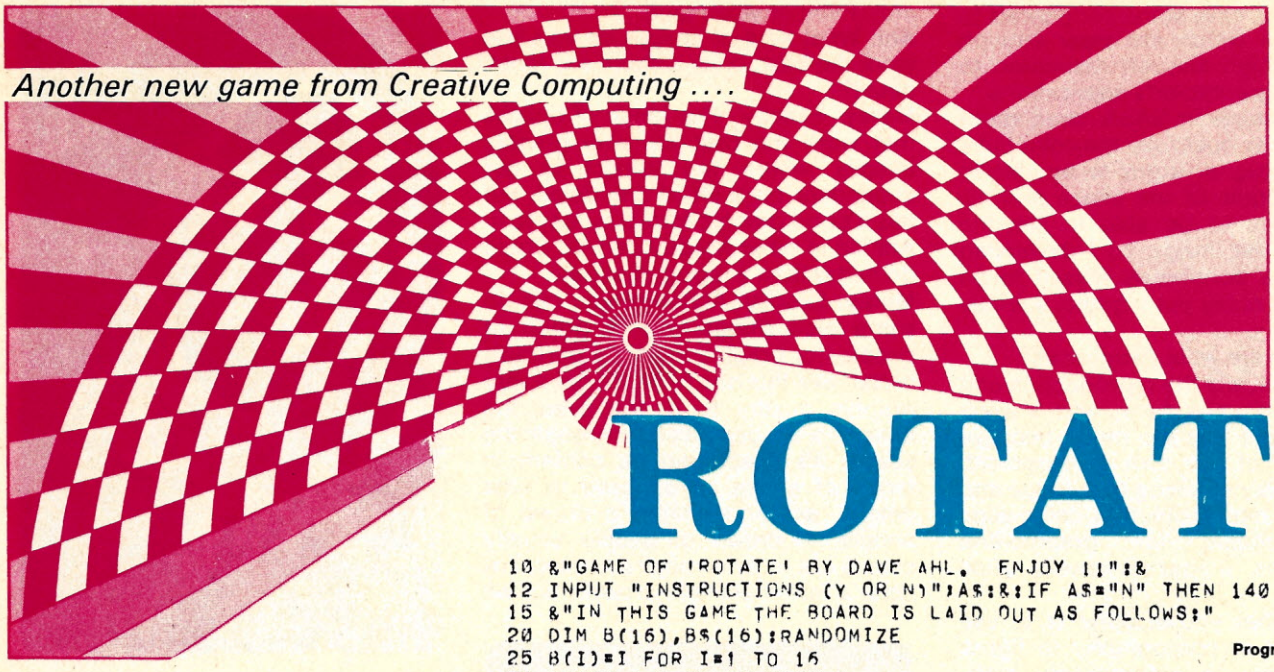
Morey: Exactly. But electronics companies and technical companies in general, like people who manufacture machinery or valves or tubes and so forth, are interested because they can take their products that have a graphic quality and make it really interesting visually and something that grabs attention. And also the electronic companies like the idea of supporting their own industry and we're a company that's done a new thing in electronics and they're applying it to a new field and they like that idea.

Ahl: How long have you been in business here?

Morey: We've been here about a year.

Ahl: In that year are you pleased with the expansion that you've had?

Morey: Yes. We're still struggling, we're still having a struggle to pay the bills, but we've gotten business from a lot of major corporations in the area and the advertising agencies and so I think we've really done quite well. But there is a lot more to do, a lot more that we can do, and I would like to see more business flowing through, I'd like to see us be able to develop more aspects of this idea than we've been able to do so far. ■



ROTATE

David H. Ahl

The game of Rotate is played on a four-by-four board filled randomly with the letters A through P. In a sense it is like the little plastic games with sliding pieces bearing the numbers 1-15 or letters A-0.

The object of the game is to put the letters in alphabetical order. This is done by rotating groups of four letters clockwise one position. The group to be rotated is specified by the positional number of the letter in the upper left-hand corner of the group. You are also given one special move which permits you to exchange any two adjacent letters. You probably don't want to use this move too early in the game; indeed, sometimes it's not necessary at all, and since you get it only one time, once you use it you can't recover. Your only move then is to type a zero to give up.

Rotate was written in DEC BASIC-PLUS and uses some peculiar symbols. In particular, the ampersand (&) is equivalent to PRINT. The CHR$ function converts from an ASCII number to a character string; for example, the statement B$=CHR$(65) would put an A in the variable B$. The alphabet (A-Z) is represented by CHR$(65) through CHR$(90). In line 325, CHR$(7) rings the Teletype bell. A FOR loop may be executed without a NEXT statement, as in the first part of line 75, which loads the letters A-P into B$(1) through B$(16). PRINT USING (line 35) prints in the fields specified by the number signs (#).

Typically, a game will take from 20 to 30 moves to win. I haven't figured out the worst possible case (assuming an intelligent method of play); I'd be happy to hear from a reader on this. Have fun!

Program Listing

```
10 &"GAME OF 'ROTATE' BY DAVE AHL.  ENJOY !!":&
12 INPUT "INSTRUCTIONS (Y OR N)";A$:&:IF A$="N" THEN 140
15 &"IN THIS GAME THE BOARD IS LAID OUT AS FOLLOWS:"
20 DIM B(16),B$(16):RANDOMIZE
25 B(I)=I FOR I=1 TO 16
30 &:FOR I=1 TO 13 STEP 4
35 PRINT USING "##  ##  ##  ##",B(I),B(I+1),B(I+2),B(I+3)
40 NEXT I:&
45 &"BOARD POSITIONS ARE OCCUPIED RANDOMLY BY THE LETTERS A TO P."
50 &"THE OBJECT OF THE GAME IS TO ORDER THE LETTERS BY ROTATING"
55 &"ANY FOUR LETTERS CLOCKWISE ONE POSITION.  YOU SPECIFY THE"
60 &"UPPER LEFT POSITION OF THE FOUR YOU WISH TO ROTATE, I.E.,"
65 &"VALID MOVES ARE 1, 2, 3, 5, 6, 7, 9, 10, AND 11."
70 &"CONSEQUENTLY, IF THE BOARD LOOKED LIKE:"
75 B$(I)=CHR$(I+64) FOR I=1 TO 16:B$(2)="C":B$(3)="G"
80 B$(6)="B":B$(7)="F":GOSUB 400
85 &"AND YOU ROTATED POSITION 2, THE BOARD WOULD BE:"
90 B$(I)=CHR$(I+64) FOR I=2 TO 7:GOSUB 400
95 &"AND YOU WOULD WIN !":&
100 &"YOU ALSO GET ONE 'SPECIAL' MOVE PER GAME WHICH YOU MAY OR"
105 &"MAY NOT NEED.  THE SPECIAL MOVE ALLOWS YOU TO EXCHANGE"
110 &"ANY TWO ADJACENT LETTERS IN A ROW.  TO MAKE THIS MOVE,"
115 &"INPUT A '-1' AS YOUR MOVE AND YOU WILL BE ASKED FOR THE"
120 &"POSITIONS OF THE TWO LETTERS TO EXCHANGE.  REMEMBER -- "
125 &"ONLY ONE SPECIAL MOVE PER GAME!":&
130 &"TO GIVE UP AT ANY TIME, TYPE A '0'.":&:&"GOOD LUCK !":&
140 B$(I)="0" FOR I=1 TO 16
150 FOR I=1 TO 16
160 T$=CHR$(INT(16*RND+65))
165 FOR J=1 TO I
170 IF B$(J)=T$ THEN 160
175 NEXT J
180 B$(I)=T$:NEXT I
190 M=0:S=0:&"HERE'S THE STARTING BOARD...":GOSUB 400
200 INPUT"POSITION TO ROTATE";I:IF I=0 THEN &:&:GOTO 140
205 IF I=-1 THEN 510
210 IF I=4 OR I=8 OR I>12 THEN &"ILLEGAL.  AGAIN...":GOTO 200
220 M=M+1:T$=B$(I)
230 B$(I)=B$(I+4):B$(I+4)=B$(I+5):B$(I+5)=B$(I+1):B$(I+1)=T$
240 GOSUB 400:REM *** PRINT BOARD
300 REM *** DO WE HAVE A WINNER?
305 FOR I=1 TO 16
310 IF CHR$(I+64)<>B$(I) THEN 200
315 NEXT I
320 &:&"YOU ORDERED THE BOARD IN"M"MOVES.":M1=M1+M:G=G+1
325 & CHR$(7);FOR I=1 TO 15
330 &:INPUT "PLAY AGAIN (Y OR N)";A$:IF A$="Y" THEN 140
340 &:&"YOU PLAYED"G"GAMES AND ORDERED THE BOARD IN AN AVERAGE"
350 &"OF"M1/G"MOVES PER GAME.":&:GOTO 999
390 REM *** PRINT BOARD SUBROUTINE
400 &:FOR I=1 TO 13 STEP 4
410 & B$(I)" "B$(I+1)" "B$(I+2)" "B$(I+3)
420 NEXT I:&:RETURN
500 REM *** SPECIAL MOVE SUBROUTINE
510 INPUT "EXCHANGE WHICH TWO POSITIONS";X,Y
520 IF X<>Y+1 AND X<>Y-1 THEN &"ILLEGAL.  AGAIN...":GOTO 510
530 S=S+1:IF S>1 THEN &"ONLY ONE SPECIAL MOVE PER GAME.":GOTO 200
540 T$=B$(X):B$(X)=B$(Y):B$(Y)=T$:GOTO 240
999 END
```

```
GAME OF 'ROTATE' BY DAVE AHL.  ENJOY !!

INSTRUCTIONS (Y OR N)? Y

IN THIS GAME THE BOARD IS LAID OUT AS FOLLOWS:

 1   2   3   4
 5   6   7   8
 9  10  11  12
13  14  15  16
```

Sample Run

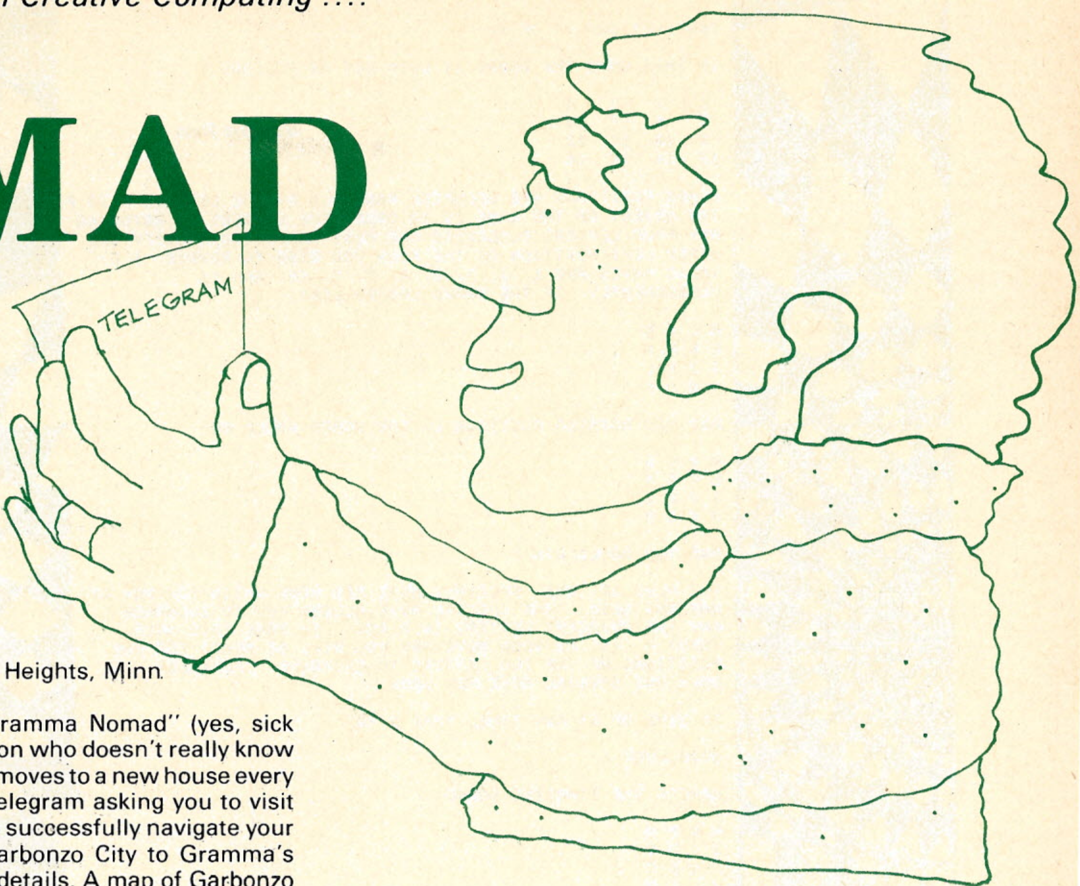
```
BOARD POSITIONS ARE OCCUPIED RANDOMLY BY THE LETTERS A TO P.
THE OBJECT OF THE GAME IS TO ORDER THE LETTERS BY ROTATING
ANY FOUR LETTERS CLOCKWISE ONE POSITION.  YOU SPECIFY THE
UPPER LEFT POSITION OF THE FOUR YOU WISH TO ROTATE, I.E.,
VALID MOVES ARE 1, 2, 3, 5, 6, 7, 9, 10, AND 11.
CONSEQUENTLY, IF THE BOARD LOOKED LIKE:

A C G D
E B F H
I J K L
M N O P

AND YOU ROTATED POSITION 2, THE BOARD WOULD BE:

A B C D
E F G H
I J K L
M N O P

AND YOU WOULD WIN !

YOU ALSO GET ONE 'SPECIAL' MOVE PER GAME WHICH YOU MAY OR
MAY NOT NEED.  THE SPECIAL MOVE ALLOWS YOU TO EXCHANGE
ANY TWO ADJACENT LETTERS IN A ROW.  TO MAKE THIS MOVE,
INPUT A '-1' AS YOUR MOVE AND YOU WILL BE ASKED FOR THE
POSITIONS OF THE TWO LETTERS TO EXCHANGE.  REMEMBER --
ONLY ONE SPECIAL MOVE PER GAME!

TO GIVE UP AT ANY TIME, TYPE A '0'.

GOOD LUCK !

HERE'S THE STARTING BOARD...

A D P N
J I O B
F G K C
H L E M

POSITION TO ROTATE? 7

A D P N
J I K O
F G C B
H L E M

POSITION TO ROTATE? 6

A D P N
J G I O
F C K B
H L E M

POSITION TO ROTATE? 7

A D P N
J G K I
F C B O
H L E M
```

----Later in the game----

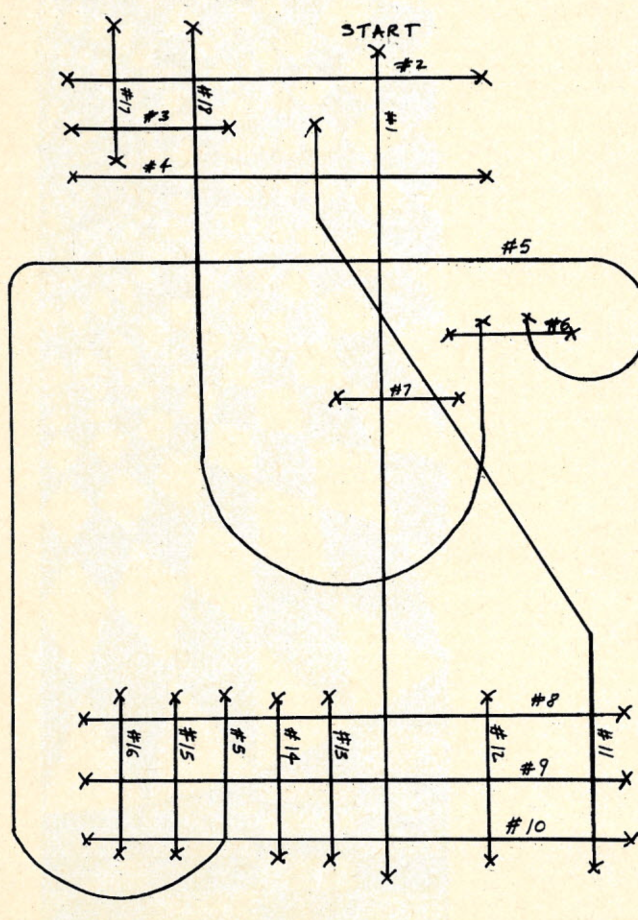
```
A B C D
E F G H
I K L P
M N J O

POSITION TO ROTATE? 11

A B C D
E F G H
I K J L
M N O P

POSITION TO ROTATE? -1
EXCHANGE WHICH TWO POSITIONS? 10,11

A B C D
E F G H
I J K L
M N O P

YOU ORDERED THE BOARD IN 26 MOVES.

PLAY AGAIN (Y OR N)? Y
```

Another new game from Creative Computing ....

NOMAD



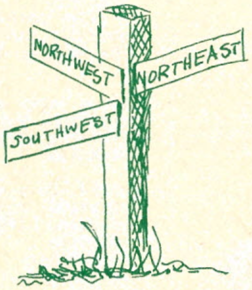
Name: Nomad I.
Author: Steve Trapp, Columbia Heights, Minn.
Language: BASIC (HP 2000F)
Description: Your gramma, "Gramma Nomad" (yes, sick joke, but it gets worse) is a person who doesn't really know where she wants to live, so she moves to a new house every game. Then she sends you a telegram asking you to visit her. The object of the game is to successfully navigate your way through the streets of Garbonzo City to Gramma's house. See the game for more details. A map of Garbonzo City is provided below for your reference.



Program Listing

```
1000 REM NOMAD I --- A GAME ****************************************
1010 REM STEVE TRAPP
1020 REM 5020 MULCARE DRIVE
1030 REM COLUMBIA HEIGHTS, MINNESOTA 55421
1040 REM (612) 571-5020
1050 REM I$[1,1] MEANS SUBSTRING OF I$ CONTAINING
1060 REM THE FIRST CHARACTER. I$[30] IN DIM STATEMENT
1070 REM MEANS I$ HAS A MAXIMUM LENGTH OF 30 CHARACTERS.
1080 REM GOTO...OF MAY BE CALLED ON...GOTO IN SOME BASIC COMPILERS.
1090 REM WRITTEN IN HEWLETT-PACKARD 2000F BASIC
1100 REM STARTED 2/11/76 -- DONE 4/28/76
1110 REM ***********************************************************
1120 REM **********************************************************
1130 REM
1140 REM
1150 REM
1160 DIM I$[30],D[2,2],R[30,30],E[30,30],P[8,2],C[30]
1170 DIM W[30]
1180 DIM N$[30]
1190 REM GOSUB RULES
1200 GOSUB 3340
1210 REM RANDOM NUMBERS
1220 DEF FNA(X)=INT(RND(0)*X)+1
1230 REM # OF ROADS
1240 READ N
1250 FOR R=1 TO N
1260 REM # OF INTERSEC
1270 READ Q
1280 C[R]=ABS(Q)
1290 IF Q<0 THEN 1320
1300 W[R]=1
1310 GOTO 1330
1320 W[R]=-1
1330 REM DIREC, ROAD
1340 FOR J=1 TO C[R]
1350 READ E[R,J],R[R,J]
1360 NEXT J
1370 NEXT R
1380 REM NAME?
1390 PRINT "WHAT IS YOUR NAME";
1400 INPUT N$
1410 REM OPENING STATEMENT
1420 PRINT
1430 PRINT "*GOOD LUCK*"
1440 REM GRAMMAS HOUSE
1450 H1=FNA(N)
1460 H2=FNA(C[H1])
1470 REM DRUNK DRIVERS
1480 MAT D=CON
1490 REM POLICE
1500 FOR A=1 TO 8
1510 P[A,1]=FNA(N)
1520 P[A,2]=FNA(C[P[A,1]])
1530 NEXT A
1540 REM LRAC NILKNARF NAMGREB LODGE
```

```
1550 R=1
1560 J=0
1570 I=1
1580 REM GOSUB CHECK
1590 GOSUB 2520
1600 REM GOSUB TELEGRAM
1610 GOSUB 2570
1620 REM ADD INCREMENT
1630 J=J+I
1640 REM NEED REPAIR?
1650 IF FNA(10)=1 THEN 2940
1660 REM DEAD END?
1670 IF J>C[R] OR J=0 THEN 3060
1680 REM DIRECTION
1690 IF I=-1 THEN 1720
1700 D=E[R,J]
1710 GOTO 1730
1720 D=9-E[R,J]
1730 REM ROAD CROSSING
1740 C=R[R,J]
1750 REM SKIP LINE
1760 PRINT
1770 REM GOSUB *DIREC, ROAD ON* PRINT
1780 GOSUB 2680
1790 REM AT GRAMMAS?
1800 IF H1=R AND R[R,J]=R[H1,H2] THEN 2880
1810 IF H1=R[R,J] AND R=R[H1,H2] THEN 2880
1820 REM JUNCTION
1830 PRINT "JUNCTION: ROAD #";R;"& #";C
1840 REM ASK WHAT WAY TO TURN
1850 PRINT "FORWARD, LEFT, RIGHT OR U-TURN";
1860 INPUT I$
1870 I$=I$[1,1]
1880 IF I$="F" THEN 2080
1890 IF I$="R" THEN 1970
1900 IF I$="L" THEN 1990
1910 IF I$="U" THEN 1950
1920 REM GOOFED
1930 PRINT "**YOU GOOFED**"
1940 GOTO 1850
1950 I=-1*I
1960 GOTO 2080
1970 I=W[R]*I
1980 GOTO 2000
1990 I=-1*W[R]*I
2000 FOR A=1 TO C[C]
2010 IF R[C,A]=R THEN 2040
2020 NEXT A
2030 STOP
2040 R=C
2050 J=A
2060 REM DARE?
2070 IF FNA(4)=1 THEN 2360
2080 REM SPEED
2090 PRINT "SPEED";
2100 INPUT S
2110 REM DANGEROUSITY CRASH CHECKS
2120 IF S>100 THEN 3190
2130 IF S<30 THEN 3220
2140 REM ILLEGAL?
2150 IF S>55 THEN 3090
2160 REM DRUNK DRIVERS DRIVE.
2170 FOR A=1 TO 2
2180 IF D[A,1]=0 THEN 2210
2190 D[A,1]=FNA(N)
2200 D[A,2]=FNA(C[D[A,1]])
2210 NEXT A
```

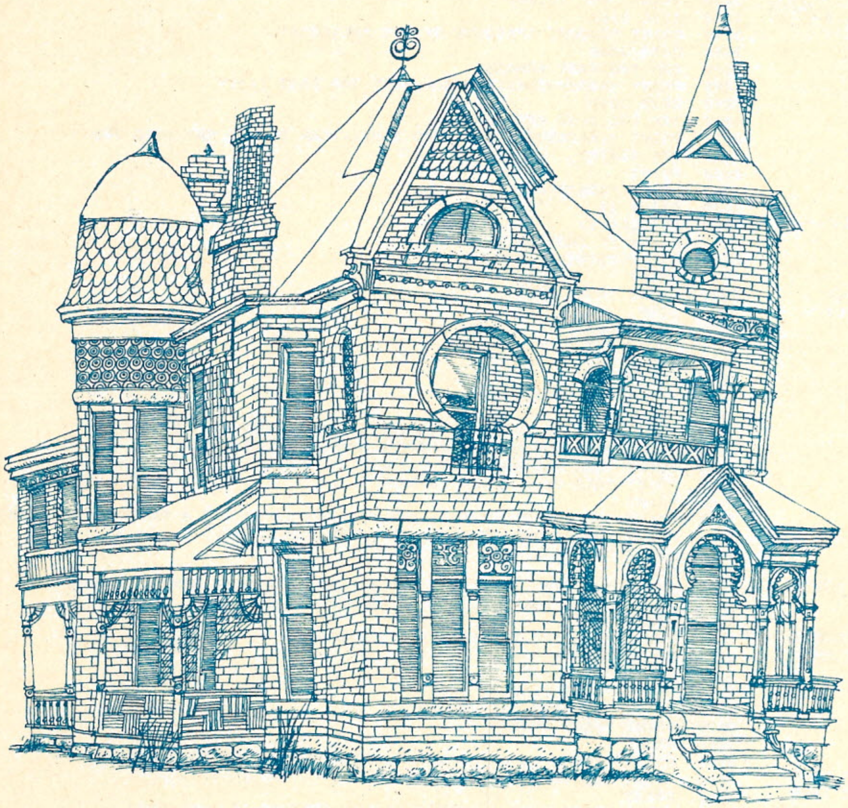



```
2220 REM HIT BY DRUNK DRIVER?
2230 FOR A=1 TO 2
2240 IF D[A,1]=0 THEN 2270
2250 IF D[A,1]=R AND R[R,J]=R[D[A,1],D[A,2]] THEN 2910
2260 IF D[A,1]=R[R,J] AND R[D[A,1],D[A,2]]=R THEN 2910
2270 NEXT A
2280 REM IS DRUNK DRIVER CAUGHT?
2290 FOR A=1 TO 2
2300 IF D[A,1]=0 THEN 2340
2310 FOR B=1 TO 8
2320 IF D[A,1]=P[B,1] AND D[A,2]=P[B,2] THEN 2440
2330 NEXT B
2340 NEXT A
2350 GOTO 1620
2360 REM SPEED DARE PRINT
2370 GOTO FNA(3) OF 2380,2400,2420
2380 PRINT "I DARE YOU TO SPEED ** (DAREDEVIL)"
2390 GOTO 2430
2400 PRINT "*SPEEDING* IS FUN  (SO DO IT)!!"
2410 GOTO 2430
2420 PRINT "*SPEED*   I DARE YOU   *SPEED*   I DARE YOU"
2430 GOTO 2080
2440 REM DRUNK CAUGHT
2450 PRINT "A DRUNK DRIVER HAS BEEN CAUGHT. THE POLICEMAN WHO"
2460 PRINT "ARRESTED HIM WILL BE TESTIFYING AT COURT FOR"
2470 PRINT "THE REST OF THE GAME."
2480 PRINT
2490 D[A,1]=D[A,2]=P[B,1]=P[B,2]=0
2500 GOTO 1620
2510 REM CHECK
2520 FOR A=1 TO 8
2530 IF H1=P[A,1] AND R[H1,H2]=R[P[A,1],P[A,2]] THEN 1440
2540 IF H1=R[P[A,1],P[A,2]] AND R[H1,H2]=P[A,1] THEN 1440
2550 NEXT A
2560 RETURN
2570 REM TELEGRAM PRINT-UP
2580 PRINT
2590 PRINT "DEAR "N$","
2600 PRINT "HOW ARE YOU? I LIVE AT THE CORNER"
2610 PRINT "OF ROAD #";H1;"& #";R[H1,H2];"!!!"
2620 PRINT "COME ON OVER."
2630 PRINT "                          LOVE,"
2640 PRINT "                          GRAMMA"
2650 PRINT "((TELEGRAMMA CORP. TELEGRAM CO.))"
2660 PRINT
2670 RETURN
2680 REM *DIREC, ROAD ON* PRINT-UP
2690 PRINT "GOING ";
2700 GOTO D OF 2710,2730,2750,2770,2790,2810,2830,2850
2710 PRINT "NORTH";
2720 GOTO 2860
2730 PRINT "WEST";
2740 GOTO 2860
2750 PRINT "NORTHEAST";
2760 GOTO 2860
2770 PRINT "SOUTHEAST";
2780 GOTO 2860
2790 PRINT "NORTHWEST";
2800 GOTO 2860
2810 PRINT "SOUTHWEST";
2820 GOTO 2860
2830 PRINT "EAST";
2840 GOTO 2860
2850 PRINT "SOUTH";
2860 PRINT " ON ROAD #";R
2870 RETURN
```

```
2880 REM AT GRAMMAS *PRINT*
2890 PRINT "YOU MADE IT TO GRAMMAS HOUSE!!!!!!!!!!"
2900 GOTO 3240
2910 REM DRUNK HIT YOUR CAR *PRINT*
2920 PRINT "KERSPLATT--DRUNK DRIVER HIT YOUR CAR."
2930 GOTO 3240
2940 REM CAR NEEDS FIXING *PRINT-UP*
2950 GOTO FNA(5) OF 2960,2980,3000,3020,3040
2960 PRINT "POP...FLAT TIRE"
2970 GOTO 3050
2980 PRINT "FLIP...YOUR CAR DID A SUMERSALT"
2990 GOTO 3050
3000 PRINT "*OUT OF GAS*"
3010 GOTO 3050
3020 PRINT "SMOOSH...BUS FLATTENED YOUR CAR."
3030 GOTO 3050
3040 PRINT "POP...BULLDOG ATE YOUR TIRE!"
3050 GOTO 3240
3060 REM DEAD END PRINT
3070 PRINT "*DEAD END*"
3080 GOTO 3240
3090 REM SPEEDING
3100 PRINT "*SPEEDING*"
3110 REM CAUGHT BY POLICE?
3120 FOR X=1 TO 8
3130 IF P[X,1]=R AND P[X,1]=J THEN 3170
3140 NEXT X
3150 PRINT "NOT CAUGHT"
3160 GOTO 2160
```

```
3170 PRINT "CAUGHT SPEEDING BY THE POLICE!!"
3180 GOTO 3240
3190 REM TOO FAST *CRASH*
3200 PRINT "KERSMOUSHIIEEEEE...WENT TOO FAST !!!!"
3210 GOTO 3240
3220 REM TOO SLOW *CRASH*
3230 PRINT "-*<(KRUNCH)>*-  TOO SLOW...CAR BEHIND RAN INTO YOU!"
3240 REM AGAIN?
3250 PRINT
3260 PRINT "AGAIN";
3270 INPUT I$
3280 I$=I$[1,1]
3290 IF I$="Y" THEN 1380
3300 REM CLOSING STATEMENT
3310 PRINT
3320 PRINT "*SEE YOU*"
3330 STOP
3340 REM RULES?
3350 PRINT "RULES";
3360 INPUT I$
3370 I$=I$[1,1]
3380 IF I$="N" THEN 3950
3390 PRINT
3400 PRINT "GRAMMA NOMAD IS A NICE OLD LADY WHO HAS NOT QUITE"
3410 PRINT "MADE UP HER MIND WHERE SHE WANTS TO LIVE."
3420 PRINT "SHE HAS NARROWED IT DOWN TO SOMEWHERE IN GARBONZO CITY"
3430 PRINT "AND ON A STREET CORNER."
3440 PRINT
3450 PRINT "AT THE BEGINNING, THE MAILMAN GIVES YOU A TELEGRAM WRITTEN"
3460 PRINT "BY GRAMMA TELLING YOU WHERE SHE LIVES."
3470 PRINT "(I WILL READ IT TO YOU)."
3480 PRINT
3490 PRINT "YOU GET INTO YOUR CAR AT LRAC NILKNARF NAMGREB LODGE."
3500 PRINT "FROM THERE YOU GO TO GRAMMAS HOUSE."
3510 PRINT
3520 PRINT "YOU TRY TO GET THERE WITHOUT:"
3530 PRINT "CRASHUPS"
3540 PRINT "TICKETS"
3550 PRINT "FLAT TIRES"
3560 PRINT "RUNNING OUT OF GAS"
3570 PRINT "DEAD ENDS"
3580 PRINT
3590 PRINT "THERE IS AN 8-MAN POLICE FORCE ENFORCING THE LAWS"
3600 PRINT "OF GARBONZO CITY."
3610 PRINT
3620 PRINT "THERE ARE 2-DRUNKS ON THE STREETS OF GARBONZO"
3630 PRINT "CITY."
3640 PRINT
3650 PRINT "IF A POLICEMAN CATCHES A DRUNK, HE HAS TO"
3660 PRINT "TESTIFY IN COURT"
3670 PRINT "(WHICH TAKES REST OF GAME)"
3680 PRINT
```

```
3690 PRINT "IF A DRUNK DRIVER HITS YOU, YOU LOSE."
3700 PRINT
3710 PRINT "IN ORDER TO FIND AND PLOT YOUR WAY TO GRAMMAS"
3720 PRINT "HOUSE, YOU NEED A MAP OF GARBONZO CITY."
3730 PRINT
3740 PRINT "TO GET A MAP, CONTACT:"
3750 PRINT
3760 PRINT "STEVE TRAPP"
3770 PRINT "5020 MULCARE DRIVE N.E."
3780 PRINT "COLUMBIA HEIGHTS, MINNESOTA 55421"
3790 PRINT "CALL (612) 571-5020."
3800 PRINT
3810 PRINT "AT EACH JUNCTION, I WILL TELL YOU:"
3820 PRINT "THE DIRECTION YOU ARE GOING"
3830 PRINT "THE ROAD YOU ARE ON"
3840 PRINT "THE ROAD CROSSING"
3850 PRINT
3860 PRINT "I WILL ASK YOU:"
3870 PRINT "THE WAY YOU WANT TO TURN (IE LEFT)"
3880 PRINT "SPEED (IN MPH)"
3890 PRINT
3900 PRINT "AN OVERPASS IS NOT A JUNCTION, SO IT IS"
3910 PRINT "MERELY SKIPPED OVER. IT IS UNANNOUNCED."
3920 PRINT
3930 PRINT "*THAT IS ALL*"
3940 PRINT
3950 RETURN
3960 REM DATA LINES
3970 DATA 18
3980 DATA -8,8,2,8,4,8,5,8,11,8,7,8,8,8,9,8,10
3990 DATA 3,7,17,7,18,7,1
4000 DATA 2,7,17,7,18
4010 DATA 3,7,18,7,11,7,1
4020 DATA 7,8,8,8,9,8,10,7,18,7,11,7,1,1,6
4030 DATA -2,7,18,7,5
4040 DATA 2,7,1,7,11
4050 DATA 8,7,16,7,15,7,5,7,14,7,13,7,1,7,12,7,11
4060 DATA 8,7,16,7,15,7,5,7,14,7,13,7,1,7,12,7,11
4070 DATA 8,7,16,7,15,7,5,7,14,7,13,7,1,7,12,7,11
4080 DATA -7,4,4,4,5,4,1,4,7,8,8,8,9,8,10
4090 DATA -3,8,8,8,9,8,10
4100 DATA -3,8,8,8,9,8,10
4110 DATA -3,8,8,8,9,8,10
4120 DATA -3,8,8,8,9,8,10
4130 DATA -3,8,8,8,9,8,10
4140 DATA -2,8,2,8,3
4150 DATA -5,8,2,8,3,8,4,8,5,1,6
4160 DATA 0,0,0,0,0,0,0
4170 REM NOMAD I WAS MADE BY STEVE TRAPP OF COLUMBIA HEIGHTS, MINNESOTA
4180 REM AS A GAME INVOLVING TURNS INSTEAD OF FUEL AND SPEED.
4190 REM ****************************************************************
4200 REM
4210 END
```



Sample Run

```
RUN
NOMAD

RULES?NO
WHAT IS YOUR NAME?SONNY

*GOOD LUCK*

DEAR SONNY,
HOW ARE YOU? I LIVE AT THE CORNER
OF ROAD # 6      & # 18    !!!
COME ON OVER.
                        LOVE,
                        GRAMMA
((TELEGRAMMA CORP. TELEGRAM CO.))


GOING SOUTH ON ROAD # 1
JUNCTION: ROAD # 1      & # 2
FORWARD, LEFT, RIGHT OR U-TURN?FORWARD
SPEED?44
A DRUNK DRIVER HAS BEEN CAUGHT. THE POLICEMAN WHO
ARRESTED HIM WILL BE TESTIFYING AT COURT FOR
THE REST OF THE GAME.


GOING SOUTH ON ROAD # 1
JUNCTION: ROAD # 1      & # 4
FORWARD, LEFT, RIGHT OR U-TURN?FORWARD
SPEED?44
*OUT OF GAS*

AGAIN?YES
WHAT IS YOUR NAME?SONNY

*GOOD LUCK*

DEAR SONNY,
HOW ARE YOU? I LIVE AT THE CORNER
OF ROAD # 9      & # 11    !!!
COME ON OVER.
                        LOVE,
                        GRAMMA
((TELEGRAMMA CORP. TELEGRAM CO.))

GOING SOUTH ON ROAD # 1
JUNCTION: ROAD # 1      & # 2
FORWARD, LEFT, RIGHT OR U-TURN?FORWARD
SPEED?44
A DRUNK DRIVER HAS BEEN CAUGHT. THE POLICEMAN WHO
ARRESTED HIM WILL BE TESTIFYING AT COURT FOR
THE REST OF THE GAME.


GOING SOUTH ON ROAD # 1
JUNCTION: ROAD # 1      & # 4
FORWARD, LEFT, RIGHT OR U-TURN?FORWARD
SPEED?55

GOING SOUTH ON ROAD # 1
JUNCTION: ROAD # 1      & # 5
FORWARD, LEFT, RIGHT OR U-TURN?FORWARD
SPEED?66
*SPEEDING*
NOT CAUGHT

GOING SOUTH ON ROAD # 1
JUNCTION: ROAD # 1      & # 11
FORWARD, LEFT, RIGHT OR U-TURN?LEFT
*SPEEDING* IS FUN  (SO DO IT)!!
SPEED?55

GOING SOUTHEAST ON ROAD # 11
JUNCTION: ROAD # 11     & # 7
FORWARD, LEFT, RIGHT OR U-TURN?F
SPEED?44

GOING SOUTH ON ROAD # 11
JUNCTION: ROAD # 11     & # 8
FORWARD, LEFT, RIGHT OR U-TURN?F
SPEED?44

GOING SOUTH ON ROAD # 11
YOU MADE IT TO GRAMMAS HOUSE!!!!!!!!!!

AGAIN?NO

*SEE YOU*

DONE
```

ANNOUNCING

Index to Advertisers

Coming in November

• **How Was I Yesterday?** With tongue in cheek and typewriter in hand, the author subtitles his article "Biorhythmic Hindsight." He provides BASIC and APL programs, and invites you to "generate a biorhythmic calendar for yourself and see if your yesterday was gloomy or great."

• **Computer-Prepared Final Exams.** In courses taught by a large number of instructors, much faculty time is spent in preparing finals. Here is a way to use computerized exams and thus save time that, as the author puts it, "could then be devoted to students and other scholarly pursuits."

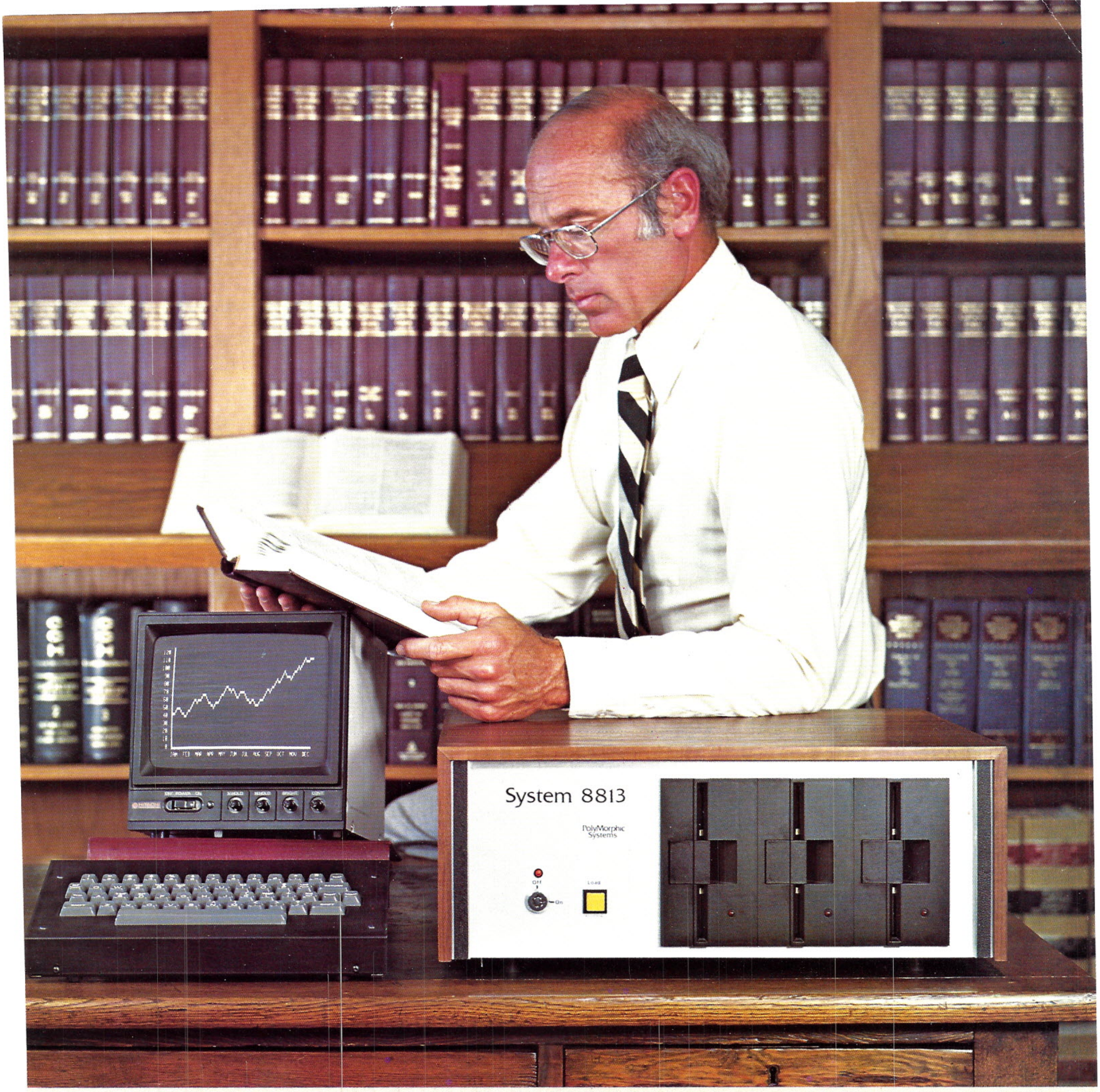
• **Something is Missing....** Craig Finseth, whose article on "A Taste of APL" in the July-August issue drew much appreciative comment, now turns his attention to dynamic, recursive interpreters. He introduces the concepts of recursion, stacks, linked lists and name tables to advanced BASIC programmers in a fascinating tutorial.

• **Games, Games, Games.** Several new ones you'll want to get up and running on your rig right away. Complete listings, runs and descriptions, naturally.

• **Holiday Special.** For your year-end enjoyment, a bouquet of impressions and misconceptions about computers as written by young children.

• **Computer-History Trivia Quiz.** What does UNIVAC stand for? Plankalkul was a 1940 attempt to do what? If you can correctly answer these and 28 other questions relating to the good old days of ENIAC, Mark I and mercury tanks, then you're a lot smarter than we are.

• **An 8-Hour Course in BASIC: Part 3.** Anyone who has read the first two parts of this series, especially Part 2 in this issue, will want to continue with this introduction to BASIC by Tom Dwyer, author of "A Guided Tour to Computer Programming in BASIC," one of the best-sellers on the *Creative Computing* book list.



The Computer for the Professional

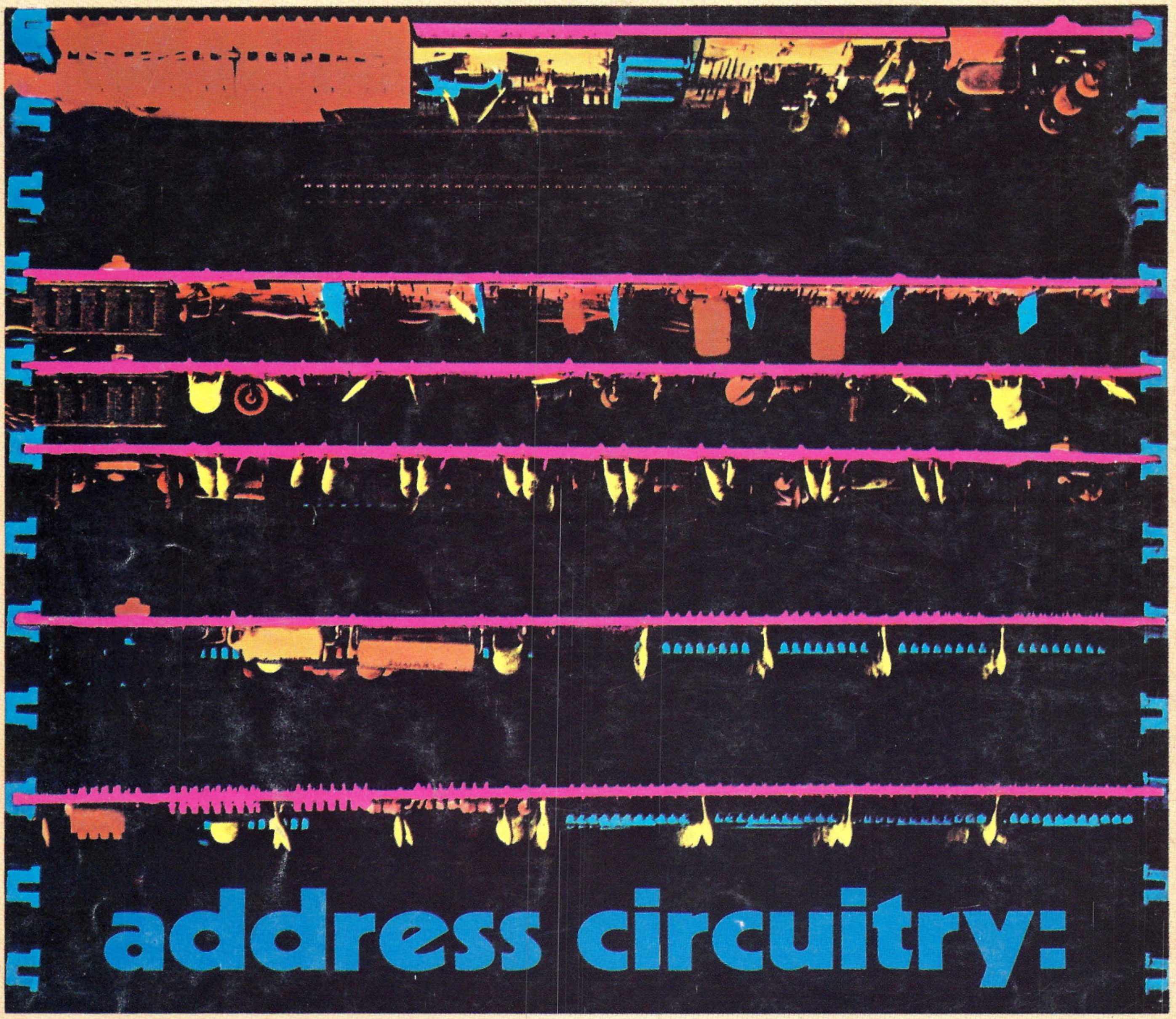
Whether you are a manager, scientist, educator, lawyer, accountant or medical professional, the System 8813 will make you more productive in your profession. It can keep track of your receivables, project future sales, evaluate investment opportunities, or collect data in the laboratory.

Use the System 8813 to develop reports, analyze and store lists and schedules, or to teach others about computers. It is easily used by novices and experts alike.

Reliable hardware and sophisticated software make this system a useful tool. Several software packages are included with the machine: an advanced disk operating system supporting a powerful BASIC language interpreter, easy to use text editor, assembler and other system utilities. Prices for complete systems start at $3250.

See it at your local computer store or contact us at 460 Ward Dr., Santa Barbara, CA 93111, (805) 967-0468.

PolyMorphic Systems



Altair Computer Centers offer you direct access to the complete line of Altair microcomputer products. For demonstrations, information or service, visit the experts at your local Altair Computer Center today. They're located at these addresses:

ALTAIR COMPUTER CENTER
4941 East 29th St.
TUCSON, AZ 85711
(602)-748-7363

COMPUTER KITS (S.F. area)
1044 University Ave.
BERKELEY, CA 94710
(415)-845-5300

THE COMPUTER STORE
(Arrowhead Computer Co.)
820 Broadway
SANTA MONICA, CA 90401
(213)-451-0713

THE COMPUTER STORE, INC.
(Hartford area)
63 South Main Street
WINDSOR LOCKS, CT 06096
(203)-627-0188

GATEWAY ELECTRONICS, INC.
OF COLORADO
2839 W. 44th Ave.
DENVER, CO 80211
(303)-458-5444

THE COMPUTER SYSTEMCENTER
3330 Piedmont Road
ATLANTA, GA 30305
(404)-231-1691

CHICAGO COMPUTER STORE
517 Talcott Rd.
PARK RIDGE, IL 60068
(312)-823-2388

THE COMPUTER STORE, INC.
120 Cambridge St.
BURLINGTON, MA 01803
(617)-272-8770

THE COMPUTER STORE
OF ANN ARBOR
310 East Washington Street
ANN ARBOR, MI 48104
(313)-995-7616

COMPUTER STORE OF DETROIT
505-507 West 11 Mile St.
MADISON HEIGHTS, MI 48071
(313)-545-2225

THE COMPUTER ROOM
3938 Beau D'Rue Drive
EAGAN, MN 55122
(612)-452-2567

GATEWAY ELECTRONICS, INC.
8123-25 Page Blvd.
ST. LOUIS, MO 63130
(314)-427-6116

ALTAIR COMPUTER CENTER
5252 North Dixie Drive
DAYTON, OH 45414
(513)-274-1149

ALTAIR COMPUTER CENTER
110 The Annex
5345 East Forty First St.
TULSA, OK 74135
(918)-664-4564

ALTAIR COMPUTER CENTER
8105 SW Nimbus Ave.
BEAVERTON, OR 97005
(503)-644-2314

ALTAIR COMPUTER CENTER
611 N. 27th St. Suite 9
LINCOLN, NB 68503
(402)-474-2800

COMPUTER STORES
OF CAROLINA, INC.
1808 E. Independence Blvd.
CHARLOTTE, N.C. 28205
(704)-334-0242

COMPUTER SHACK
3120 San Mateo N.E.
ALBUQUERQUE, NM 87110
(505)-883-8282, 883-8283

THE COMPUTER STORE
269 Obsorne Rd.
ALBANY, NY 12211
(518)-459-6140

THE COMPUTER STORE
OF NEW YORK
55 West 39th St.
NEW YORK, NY 10018
(212)-221-1404

ALTAIR COMPUTER CENTER
3208 Beltline Road
Suite 206
DALLAS, TX 75234
(214)-241-4088 Metro-263-7638

ALTAIR COMPUTER CENTER
5750 Bintliff Drive
HOUSTON, TX 77036
(713)-780-8981

COMPUTERS-TO-GO
4503 West Broad St.
RICHMOND, VA 23230
(804)-335-5773

MICROSYSTEMS (Washington, D.C.)
6605A Backlick Rd.
SPRINGFIELD, VA 22150
(703)-569-1110

THE COMPUTER STORE
Suite 5
Municipal Parking Building
CHARLESTON, W.VA. 25301
(304)-345-1360

mits